1990-12

Multiple input sliding mode control for autonomous diving and steering of underwater vehicles

Hawkinson, Todd D.

Monterey, California: Naval Postgraduate School

(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MULTIPLE INPUT SLIDING MODE CONTROL FOR AUTONOMOUS DIVING AND STEERING OF UNDERWATER VEHICLES

by

Todd D. Hawkinson

December, 1990

Thesis Advisor: Fotis A. Papoulias

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL ME (If applicable)	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
MULTIPLE INPUT SLIDING MODE CONTROL FOR AUTONOMOUS DIVING AND STEERING OF UNDERWATER VEHICLES

12 PERSONAL AUTHOR(S)
Hawkinson, Todd D.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Mechanical Engineer	FROM _____ TO _____	December 1990	214

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Autonomous, Underwater Vehicles, AUV, Guidance Control

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Design and analysis of multiple input autopilots using sliding modes in order to achieve accurate horizontal and vertical plane control of an autonomous underwater vehicle over a wide variation of speeds is presented. The simulated vehicle is equipped with two (fore and aft) sets of dive planes and two sets of rudders. In addition, two vertical and two horizontal thrusters are provided for control during low speed or hovering operations. The entire range of vehicle speeds from zero speed hovering to full speed ahead is divided into regions depending on control efficiency. Thrusters are used for low speed hovering, control surfaces for transition speeds. Linear quadratic regulator optimal control techniques coupled with the robustness properties of sliding mode control are utilized to provide the necessary control reversal which occurs during the transition from cruise to hover mode. Constant disturbances arising from underwater currents are effectively compensated resulting in

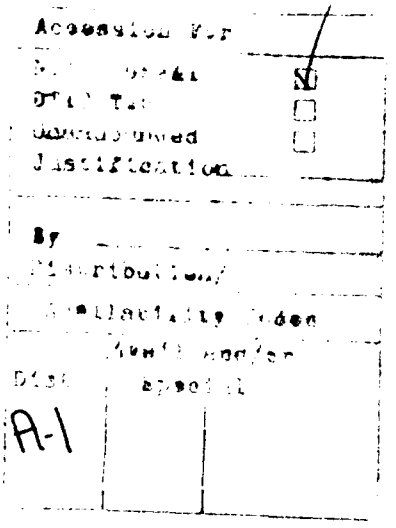
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Fotis A. Papoulias	22b TELEPHONE (Include Area Code) (408) 646-3381 22c OFFICE SYMBOL MEPa

DD Form 1473, JUN 86 *Previous editions are obsolete*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

S/N 0102-LF-014-6603

(19) Continued:
accurate path keeping. As a consequence of the multiple input control methodology developed in this work, it is shown that both path and orientation accuracy can be achieved in moderate cross current environments. Finally, reduced order observers are designed in order to account for sensor absence or malfunction.



Approved for public release; distribution is unlimited.

Multiple Input Sliding Mode Control for
Autonomous Diving and Steering of
Underwater Vehicles

by

Todd D. Hawkinson
Lieutenant, United States Navy
B.S., University of Minnesota

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
and
MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1990

Author: [redacted]
Todd D. Hawkinson

Approved by: [redacted]
Fotis A. Papoulias, Thesis Advisor

[redacted]
Anthony J. Healey, Chairman
Department of Mechanical Engineering

[redacted]
DEAN OF FACULTY
AND GRADUATE STUDIES

ABSTRACT

Design and analysis of multiple input autopilots using sliding modes in order to achieve accurate horizontal and vertical plane control of an autonomous underwater vehicle over a wide variation of speeds is presented. The simulated vehicle is equipped with two (fore and aft) sets of dive planes and two sets of rudders. In addition, two vertical and two horizontal thrusters are provided for control during low speed or hovering operations. The entire range of vehicle speeds from zero speed hovering to full speed ahead is divided into regions depending on control efficiency. Thrusters are used for low speed hovering, control surfaces for high speed cruising, and a combination of thrusters and control surfaces for transition speeds. Linear quadratic regulator optimal control techniques coupled with the robustness properties of sliding mode control are utilized to provide the necessary control reversal which occurs during the transition from cruise to hover mode. Constant disturbances arising from underwater currents are effectively compensated resulting in accurate path keeping. As a consequence of the multiple input control methodology developed in this work, it is shown that both path and orientation accuracy can be achieved in moderate cross current environments. Finally, reduced order observers are designed in order to account for sensor absence or malfunction.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. GENERAL 1

B. AIM OF THIS STUDY 2

C. THESIS OUTLINE 3

II. THEORY OF MIMO SLIDING MODE CONTROL 5

A. INTRODUCTION 5

B. REVIEW OF SLIDING MODE THEORY 5

1. Overview of MIMO Sliding Mode Control Equations 5

2. Liapunov Function 7

3. Determination of the Control Laws 8

4. Determination of the Sliding Surfaces 9

C. ALTERNATIVE APPROACH 10

1. Transformations 10

a. Transformation Matrix 10

b. Transformed Equations of Motion 11

c. Transformed Sliding Surfaces 12

d. Transformed Liapunov Function 13

e. Transformed Control Laws 13

2. Transformed Equations on the Sliding Surfaces 14

3. LQR Method for Determination of the Sliding Surfaces 14

4. Determination of the Control Laws 17

D. CONTROL OF CHATTERING 17

E. COMMENTS 18

III. MIMO SLIDING MODE CONTROL OF THE AUV IN THE DIVE PLANE 19

A. NONLINEAR GOVERNING EQUATIONS IN THE DIVE PLANE 19

1. Modifications to the SDV Equations of Motion 19

a. Tunnel Thrusters 19

2. Identification of Symbols 20

3. Assumptions 22

4. Resulting Equations 23

B. CONTROL METHODOLOGY 25

C. CONTROL OF THE AUV IN THE HOVER REGION 28

1. State Space Equations 28

2. Controller Design by Pole Placement 29

3. Computer Simulation with Pole Placement Controller 31

4. Controller Design by LQR Technique 34

5. Computer Simulation with LQR Controller 35

D. CONTROL OF THE AUV IN THE CRUISE REGION 40

1. State Space Equations 40

2. Controller Design by Pole Placement 40

3. Computer Simulation with Pole Placement Controller 41

4. Controller Design by LQR Technique 44

5. Computer Simulation with LQR Controller 44

E. CONTROL OF THE AUV IN THE TRANSITION REGION 49

1. State Space Equations 49

2. Controller Design by LQR Technique 51

3. Computer Simulation with LQR Controller 51

a. Modified LQR Controller 52

F. OBSERVERS 61

1. Design 61

2. Simulation with Observers 64

G. TRANSITION IN/OUT OF HOVER 71

H. CONCLUDING REMARKS 71

IV. LINE-OF-SIGHT GUIDANCE OF THE AUV USING MIMO SLIDING MODE CONTROL 77

A. INTRODUCTION 77

B. NONLINEAR GOVERNING EQUATIONS IN THE HORIZONTAL PLANE 78

1. Addition of Bow Rudder to SDV Model 78

2. Assumptions 78

3. Resulting Equations 79

C. LINEAR STATE SPACE REPRESENTATION 81

D. CONTROLLER DESIGN BY LQR TECHNIQUE 82

E. LINE-OF-SIGHT GUIDANCE SCHEME 83

1. Hit Criterion 83

2. Shortest Turn 83

F. SIMULATION 84

G. STEADY-STATE DISTURBANCE COMPENSATION 87

1. Theory 87

2. AUV Simulation with Steady-State Disturbance Compensation 94

H. OBSERVERS 97

1. Design 97

2. Simulation with Observers 99

I. CONCLUDING REMARKS 104

V. CROSS-TRACK-ERROR GUIDANCE OF THE AUV USING MIMO SLIDING MODE CONTROL 105

A. INTRODUCTION 105

B. CROSS-TRACK-ERROR GUIDANCE SCHEME 105

C. CONTROLLER DESIGN BY LQR TECHNIQUE 106

D. STEADY-STATE DISTURBANCE COMPENSATION 107

1. Theory of Heading Error Compensation 107

2. Simulation with Heading Error Compensation 111

3. Theory of Rudder Action Compensation 116

4. Simulation with Rudder Action Compensation 118

F. CONCLUDING REMARKS 128

VI. AUV SIMULATION IN THREE DIMENSIONAL SPACE 130

A. INTRODUCTION 130

B. SIMULTANEOUS LOS STEERING AND DEPTH CONTROL . 130

C. SIMULTANEOUS CTE STEERING AND DEPTH CONTROL . 133

D. THREE DIMENSIONAL VEHICLE PATH KEEPING 133

1. Path Keeping in the Vertical Plane 136

2. Three Dimensional Path Keeping 145

E. CONCLUDING REMARKS 150

VII. CONCLUSIONS AND RECOMMENDATIONS 155

A. CONCLUSIONS 155

B. RECOMMENDATIONS 155

APPENDIX A. SUMMARY LISTING OF COMPUTER PROGRAMS ... 157

APPENDIX B. MATRIX-x LQR DESIGN PROGRAM FOR MIMO SLIDING MODE CONTROLLERS 161

APPENDIX C. AUV DIVING SIMULATION PROGRAM; MIMO SLIDING MODE CONTROLLER DESIGNED BY LQR METHOD; W OBSERVED 162

APPENDIX D. AUV STEERING SIMULATION PROGRAM; MIMO SLIDING MODE LOS CONTROLLER DESIGNED BY LQR METHOD; DISTURBANCE REJECTION; V, U_C, AND V_C OBSERVED 173

APPENDIX E. AUV STEERING SIMULATION PROGRAM; MIMO SLIDING MODE CTE CONTROLLER DESIGNED BY LQR METHOD; DISTURBANCE REJECTION; V, U_C, AND V_C OBSERVED 184

LIST OF REFERENCES 195

INITIAL DISTRIBUTION LIST 196

LIST OF TABLES

Table 1. DEFINITION OF AUV STATES 21

Table 2. DEFINITION OF AUV CONTROLS 21

Table 3. DEFINITION OF ADDITIONAL VARIABLES 22

LIST OF FIGURES

Figure 1. Forces Produced by Planes and Thrusters vs Surge Velocity 27

Figure 2. Run 1 - AUV Response in the Dive Plane at Hover Speed, Pole Placement (u = 0.25 ft/s, ϕ = 2.0, η^2 = 0.2, PSF) 32

Figure 3. Run 1 - (continued) 33

Figure 4. Run 2 - AUV Response in the Dive Plane at Hover Speed, LQR (u = 0.00 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback) 36

Figure 5. Run 2 - (continued) 37

Figure 6. Run 3 - AUV Response in the Dive Plane at Hover Speed, LQR (u = 0.55 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback) 38

Figure 7. Run 3 - (continued) 39

Figure 8. Run 4 - AUV Response in the Dive Plane at Cruise Speed, Pole Placement (u = 3.875 ft/s, ϕ = 1.0, η^2 = 0.35, PSF) 42

Figure 9. Run 4 - (continued) 43

Figure 10. Run 5 - AUV Response in the Dive Plane at Cruise Speed, LQR (u = 1.90 ft/s, ϕ = 1.0, η^2 = 0.35, Perfect State Feedback) 45

Figure 11. Run 5 - (continued) 46

Figure 12. Run 6 - AUV Response in the Dive Plane at Cruise Speed, LQR (u = 6.00 ft/s, ϕ = 1.0, η^2 = 0.35, Perfect State Feedback) 47

Figure 13. Run 6 - (continued) 48

Figure 14. Run 7 - AUV Response in Dive Plane at Transition Speed, LQR (u = 0.90 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback) 53

Figure 15. Run 7 - (continued) 54

Figure 16. Run 8 - AUV Response in Dive Plane at Transition Speed, LQR (u = 1.60 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback) 55

Figure 17. Run 8 - (continued) 56

Figure 18. Run 9 - Diving Response at Transition Speed, MODIFIED LQR (u = 0.90 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback) 57

Figure 19. Run 9 - (continued) 58

Figure 20. Run 10 - Diving Response at Transition Speed, MODIFIED LQR (u = 1.60 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback) 59

Figure 21. Run 10 - (continued) 60

Figure 22. Run 11 - AUV Response in the Dive Plane at Hover Speed, LQR (u = 0.00 ft/s, ϕ = 2.0, η^2 = 0.2, w Observed) 65

Figure 23. Run 11 - (continued) 66

Figure 24. Run 12 - AUV Response in the Dive Plane at Cruise Speed, LQR (u = 6.00 ft/s, ϕ = 1.0, η^2 = 0.35, w Observed) 67

Figure 25. Run 12 - (continued) 68

Figure 26. Run 13 - AUV Response in Dive Plane at Transition Speed, LQR (u = 0.90 ft/s, ϕ = 2.0, η^2 = 0.2, w Observed) 69

Figure 27. Run 13 - (continued) 70

Figure 28. Run 14 - AUV Response While Transitioning to Hover, LQR (ϕ = 2.0, η^2 = 0.2, w Observed) 72

Figure 29. Run 14 - (continued) 73

Figure 30. Run 15 - AUV Response While Transitioning Out of Hover, LQR (ϕ = 2.0, η^2= 0.2, w Observed) 74

Figure 31. Run 15 - (continued) 75

Figure 32. Fortran Code for Calculating the Shortest Turn 84

Figure 33. Run 16 - AUV Steering Response in the Horizontal Plane, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, PSF, No Current) 85

Figure 34. Run 16 - (continued) 86

Figure 35. Run 17 - AUV Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, PSF, u_{co} = v_{co} = 2.0 ft/s, No Disturbance Compensation) 88

Figure 36. Run 17 - (continued) 89

Figure 37. Run 18 - AUV Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, PSF, u_{co} = v_{co} = 2.0 ft/s, Disturbance Compensation) . . . 95

Figure 38. Run 18 - (continued) 96

Figure 39. Run 19 - Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, Observers, u_{co} = v_{co} = 2.0 ft/s, Disturbance Compensation) 100

Figure 40. Run 19 - (continued) 101

Figure 41. Run 20 - Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, Observers, u_{co} = v_{co} = 2.0 ft/s, Disturbance Compensation) 102

Figure 42. Run 20 - (continued) 103

Figure 43. Run 21 - Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, Hit = 0.5L) 112

Figure 44. Run 21 - (continued) 113

Figure 45. Run 22 - Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, Hit = 7L) 114

Figure 46. Run 22 - (continued) 115

Figure 47. Run 23 - Steering Response with CTE Guidance, (Disturbance Compensation by Rudder Action, v_c = -1.0 ft/s) 119

Figure 48. Run 23 - (continued) 120

Figure 49. Run 24 - Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, v_c = -1.0 ft/s) 122

Figure 50. Run 24 - (continued) 123

Figure 51. Run 25 - Steering Response with CTE Guidance, (Disturbance Compensation by Rudder Action, v_c = -2.0 ft/s) 124

Figure 52. Run 25 - (continued) 125

Figure 53. Run 26 - Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, v_c = -2.0 ft/s) 126

Figure 54. Run 26 - (continued) 127

Figure 55. Run 27 - AUV Simulation During Simultaneous Course and Depth Changes, LOS Steering Controller, Perfect State Feedback 131

Figure 56. Run 27 - (continued) 132

Figure 57. Run 28 - AUV Simulation During Simultaneous Course and Depth Changes, CTE Steering Controller, Perfect State Feedback 134

Figure 58. Run 28 - (continued) 135

Figure 59. Coordinate System in the Vertical Plane 137

Figure 60. Path Keeping in the Vertical Plane with Stern Planes Only (z_G = 0.1 ft (nominal), z_G = 0.3 ft, and z_G = 0.5 ft) 143

Figure 61. Runs 29 and 30 - Comparison of Path Keeping in the Vertical Plane with Stern Planes Only (Curves 1) and Stern & Bow Planes (Curves 2) 146

Figure 62. Runs 29 and 30 - (continued) 147

Figure 63. Three Dimensional Geometry and Related Nomenclature 148

Figure 64. Coordinate Rotations in Three Dimensional Space 149

Figure 65. Run 31 - Three Dimensional Path Keeping at u = 3.0 ft/s 151

Figure 66. Run 32 - Three Dimensional Path Keeping at u = 2.0 ft/s 152

Figure 67. Run 33 - Three Dimensional Path Keeping at u = 1.5 ft/s 153

ACKNOWLEDGMENTS

I would like to express sincere gratitude to my thesis adviser, Professor Fotis A. Papoulias. His expert assistance and knowledgeable advice allowed me to overcome many intractable difficulties.

No acknowledgement would be complete, however, without recognition of the contributions made by my wife, Louise, and my daughter, Brenda. Their unselfish sacrifices enabled me to concentrate on the completion of this thesis to the exclusion of my family responsibilities. Their support truly made this thesis possible. I hope to repay the debt in the future.

I. INTRODUCTION

A. GENERAL

The research area of automatic control of autonomous underwater vehicles, or AUVs, is of particular interest to the Navy and private industry. AUVs are capable of a variety of unclassified missions including ASW, decoy, survey, reconnaissance, and ocean engineering work. The attractiveness of small unmanned vehicles is increasing due to the escalating capital and operational costs of manned submarines.

The dynamics of underwater vehicles are described by highly nonlinear systems of equations with uncertain coefficients and disturbances that are difficult to measure. An automatic controller for an AUV must satisfy two conflicting requirements: First, it must be sophisticated enough to perform its mission in an open ocean environment with ever-changing vehicle/environment interactions. Second, it must be simple enough to achieve real-time control without nonessential computational delays.

Sliding mode control theory yields a design that fulfills the above requirements. It provides accurate control of nonlinear systems despite unmodeled system dynamics and disturbances. Furthermore, a sliding mode controller is easy to design and implement. A very effective pseudo-linear sliding mode controller can be developed from the linearized equations of motion for an underwater vehicle.

The sliding mode control concept was first introduced by Utkin [Ref. 1] and recently developed by Slotine [Ref. 2]. At the U.S. Naval Postgraduate School (NPS),

the sliding mode theory has been applied by several past students for control of an AUV, e.g., Lienard [Ref. 3] and Joo-No Sur [Ref. 4]. However, past students have approached the problem of AUV control as a single input/single output (SISO) system. So far, satisfactory control has been realized with this method, but improved control should be achievable by applying multiple input/multiple output (MIMO) sliding mode control theory. In fact, it is imperative to treat an AUV as a MIMO system to be able to control it through the entire speed range from hovering at zero speed to maximum speed.

B. AIM OF THIS STUDY

The aim of this study is to develop a robust MIMO sliding mode controller that will control speed, heading, and depth through a wide range of operational speeds even in the presence of underwater currents.

An AUV has been recently constructed at NPS and is currently being outfitted. The hydrodynamic coefficients of the NPS AUV are unknown, hence a surrogate vehicle with known characteristics is needed to explore control theory. As in prior AUV control theses at NPS, the Mark IX Swimmer Delivery Vehicle (SDV) will be utilized.

The first step of this study is the development of the linear quadratic regulator (LQR) optimal control technique for MIMO sliding mode controller design. This method is utilized for the design of a depth controller. In the vertical plane, the full nonlinear equations of motion are simplified until they become decoupled from the

horizontal plane equations. The MIMO sliding mode depth controller is designed and simulated based on these simplified nonlinear equations.

The LQR method is then utilized for the design of a line-of-sight (LOS) steering controller and then a cross-track-error (CTE) steering controller. These controllers are also designed and simulated based on simplified nonlinear equations of motion.

The diving and steering (first the LOS and then the CTE) controllers are then combined into a control package. This package is then used to control the AUV utilizing the full nonlinear equations of motion.

Finally, three dimensional path keeping is developed. This method uses two cross-track-error controllers to allow the AUV to follow a straight line path in three dimensional space.

In all cases, reduced order observers are designed to estimate states that are not directly measurable. Additionally, the speed controller developed by Lienard [Ref. 3] was utilized in all cases.

C. THESIS OUTLINE

In Chapter II, the LQR method of MIMO sliding mode controller design is developed. Chapter III discusses the diving controller design and simulation. Chapter IV is concerned with the LOS steering controller. In Chapter V, the CTE steering controller is developed. Simulation of the AUV with the full nonlinear equations of motion and the combined control package is achieved in Chapter VI.

In addition, three dimensional path keeping is developed and simulated in Chapter VI. Finally, Chapter VII contains the conclusions and recommendations.

II. THEORY OF MIMO SLIDING MODE CONTROL

A. INTRODUCTION

This chapter is devoted to the theory of MIMO sliding mode control. A more thorough discussion of sliding mode theory can be found in [Ref. 1] and [Ref. 2]. In this chapter, sliding mode theory is reviewed with emphasis on the differences due to MIMO applications. Then, general procedures for designing a sliding mode MIMO controller are delineated.

B. REVIEW OF SLIDING MODE THEORY

1. Overview of MIMO Sliding Mode Control Equations

The controller design process begins with the linearized equations of motion written in the standard state space representation as follows:

$$\dot{x} = Ax + Bu \tag{2.1}$$

where, $x \in \mathbf{R}^n$, state vector

$u \in \mathbf{R}^m$, control input vector

$A \in \mathbf{R}^{n \times n}$, state matrix

$B \in \mathbf{R}^{n \times m}$, control distribution matrix

n, number of states

m, number of inputs

The sliding mode control laws, u, for the system of (2.1) are of the form:

$$u = \hat{u} + \bar{u} \tag{2.2}$$

As can be seen in (2.2), the control laws are composed of two parts. It will be seen later that the first, $\hat{u}$, are linear feedback laws based on the nominal linearized model (2.1). The second, $\bar{u}$, are nonlinear feedbacks with their signs switching between plus and minus according to the location of the system with respect to the sliding surfaces:

$$\sigma(x) = S^T x = 0 \tag{2.3}$$

where, $S \in \mathbf{R}^{n \times m}$

$$\sigma \in \mathbf{R}^m$$

Determination of S will determine the sliding surfaces uniquely.

The control laws (2.2) must be able to drive the system (2.1) onto the sliding surfaces (2.3) and the nominal operation point for an arbitrary choice of initial conditions. It will be seen later that the dimension of the m sliding surfaces must be one less than the dimension of the state space since $\bar{u}$ has to change sign as the system crosses $\sigma(x) = 0$. Detailed development of $\hat{u}$ and $\bar{u}$ is in the sections that follow. [Ref. 5]

2. Liapunov Function

By defining the Liapunov function:

$$V(\boldsymbol{x}) = \frac{1}{2}[\boldsymbol{\sigma}(\boldsymbol{x})]^2 = \frac{1}{2}(\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_m^2) \tag{2.4}$$

asymptotic stability of (2.3) is guaranteed provided $\dot{V}(\boldsymbol{x})$ is a negative definite function:

$$\dot{V}(\boldsymbol{x}) = \sigma_1 \dot{\sigma}_1 + \sigma_2 \dot{\sigma}_2 + \ldots + \sigma_m \dot{\sigma}_m < 0 \tag{2.5}$$

This is true if:

$$\begin{aligned} \sigma_1 \dot{\sigma}_1 &< 0 \\ \sigma_2 \dot{\sigma}_2 &< 0 \\ &\vdots \\ \sigma_m \dot{\sigma}_m &< 0 \end{aligned} \tag{2.6}$$

which can be written in the form:

$$\begin{aligned} \dot{\sigma}_1 &= -\eta_1^2 \, sign(\sigma_1) \\ \dot{\sigma}_2 &= -\eta_2^2 \, sign(\sigma_2) \\ &\vdots \\ \dot{\sigma}_m &= -\eta_m^2 \, sign(\sigma_m) \end{aligned} \tag{2.7}$$

3. Determination of the Control Laws

Using (2.1) and (2.3) in (2.7) results in:

$$S^T(Ax + Bu) = -k_n \tag{2.8}$$

where, $k_n = \begin{Bmatrix} \eta_1^2 \, sign(\sigma_1) \\ \eta_2^2 \, sign(\sigma_2) \\ \cdot \\ \cdot \\ \cdot \\ \eta_m^2 \, sign(\sigma_m) \end{Bmatrix}$

Solving for u:

$$u = -(S^T B)^{-1} S^T Ax - (S^T B)^{-1} k_n \tag{2.9}$$

which is the same form as (2.2) with:

$$\hat{u} = -(S^T B)^{-1} S^T Ax \tag{2.10}$$

$$\bar{u} = (S^T B)^{-1} k_n \tag{2.11}$$

When on the sliding surfaces $\sigma(x) = 0$, the control laws become $u = \hat{u}$. Substituting (2.10) into the system (2.1) yields:

$$\dot{x} = [A - B(S^T B)^{-1} S^T A]x \tag{2.12}$$

which can be written:

$$\dot{x} = (A - Bk)x \tag{2.13}$$

where, $\boldsymbol{k} = (\boldsymbol{S}^T \boldsymbol{B})^{-1} \boldsymbol{S}^T \boldsymbol{A}$

The gain matrix, $\boldsymbol{k}$, can be found from standard pole placement or LQR methods. The closed loop dynamics matrix:

$$\boldsymbol{A}_c = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{k} \tag{2.14}$$

has eigenvalues specified for desirable response. Additionally, m poles of $\boldsymbol{A}_c$ must be zero which is consistent with the decomposition (2.9).

The linear feedbacks $\hat{\boldsymbol{u}}$ provide the desired dynamics on the sliding surfaces only. Therefore, $\hat{\boldsymbol{u}}$ has no effect for the m excursions off the sliding surfaces $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{0}$.

Conversely, the nonlinear feedbacks $\bar{\boldsymbol{u}}$ only drive the system onto the sliding surfaces and provide no control action on the sliding surfaces. The one requirement for this action is the gains η_i^2 have to be chosen large enough so $\bar{\boldsymbol{u}}$ can provide the required robustness due to momentary disturbances and unmodeled dynamics without any compromise in stability.

4. Determination of the Sliding Surfaces

With $\boldsymbol{A}_c$ and $\boldsymbol{k}$ specified, $\boldsymbol{S}$ can be determined as follows. Rearranging the expression for $\boldsymbol{k}$ in (2.13) and using the expression for $\boldsymbol{A}_c$ in (2.14) yields:

$$\boldsymbol{S}^T \boldsymbol{A}_c = \boldsymbol{0} \tag{2.15}$$

The existence of m linearly independent nontrivial solutions of (2.15) is guaranteed if the closed loop dynamic matrix has rank deficiency m, i.e., if A_c has m poles at the origin. This is consistent with the decomposition (2.9). The linear feedbacks $\hat{u}$ provide the desired dynamics on the m sliding surfaces only. Therefore, $\hat{u}$ has no effect in the directions that are perpendicular to the sliding planes. However, it is not always possible to compute S from (2.15) directly since the existence of m left eigenvectors of A_c corresponding to the multiplicity m zero eigenvalue is not always guaranteed. Thus, another method is required for MIMO sliding mode controller design. This leads us to a method proposed by Utkin.

C. ALTERNATIVE APPROACH

In [Ref. 6], Utkin proposed an alternative method of sliding mode controller design especially for MIMO applications. This method requires a transformation of the state vector and associated matrices.

1. Transformations

a. Transformation Matrix

First, an orthogonal nxn transformation matrix, T, is found such that:

$$TB = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tag{2.16}$$

where, $B_1 \in \mathbf{R}^{m \times m}$

$0 \in \mathbf{R}^{(n-m) \times m}$

QR factorization which is available on software programs such as Matrix-x (Copyright 1989 by Integrated Systems Inc.) can be used to determine T. After QR factorization, the B matrix is transformed to:

$$B = Q\begin{bmatrix} R \\ 0 \end{bmatrix} \tag{2.17}$$

where Q is orthogonal and R is upper triangular. Rearranging (2.17) yields:

$$Q^{T}B = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{2.18}$$

where then, $T = Q^{T}$

$$B_1 = R$$

Note that the transformation need not be applied if the B matrix is already of the desired form of (2.16). This was found to be the situation for all cases of MIMO control of the SDV.

b. Transformed Equations of Motion

After the transformation, the new states are:

$$y = Tx \tag{2.19}$$

and the system (2.1) becomes:

$$\dot{y} = TAT^{T}y + TBu \tag{2.20}$$

which can be decomposed into the form:

$$\dot{y}_1 = A_{11} y_1 + A_{12} y_2 + B_1 u \tag{2.21}$$

$$\dot{y}_2 = A_{21} y_1 + A_{22} y_2 \tag{2.22}$$

where, $y_1 \ \varepsilon \ \mathbf{R}^{m}$

$y_2 \ \varepsilon \ \mathbf{R}^{(n-m)}$

$A_{11} \ \varepsilon \ \mathbf{R}^{mxm}$

$A_{12} \ \varepsilon \ \mathbf{R}^{mx(n-m)}$

$A_{21} \ \varepsilon \ \mathbf{R}^{(n-m)xm}$

$A_{22} \ \varepsilon \ \mathbf{R}^{(n-m)x(n-m)}$

$B_1 \ \varepsilon \ \mathbf{R}^{mxm}$

c. Transformed Sliding Surfaces

Using the transformation of the states (2.19) in the equation of the sliding surfaces (2.3) results in:

$$\sigma(y) = C^{T} y = 0 \tag{2.23}$$

where, $C = TS$

Equation (2.23) can be decomposed to:

$$\sigma(y) = C_1^{T} y_1 + C_2^{T} y_2 = 0 \tag{2.24}$$

where, $C_1^{T} \ \varepsilon \ \mathbf{R}^{mxm}$

$$C_2^T \in \mathbb{R}^{m \times (n-m)}$$

$$0 \quad \in \mathbb{R}^m$$

Without loss of generality, the matrix C_1 can be set equal to the identity matrix I to simplify the equations for the sliding surfaces.

d. Transformed Liapunov Function

As before, a Liapunov function is defined to ensure asymptotically stable sliding surfaces. In transformed states, the Liapunov function is:

$$V(y) = \frac{1}{2}[\sigma(y)]^2 \tag{2.25}$$

which again leads to (2.7).

e. Transformed Control Laws

Substituting (2.24) into (2.7) leads to the following expression:

$$\dot{y}_1 + C_2^T \dot{y}_2 = -k_n \tag{2.26}$$

Substituting (2.21) and (2.22) into (2.26) and solving for u gives:

$$u = -B_1^{-1}[(A_{11} + C_2^T A_{21})y_1 + (A_{12} + C_2^T A_{22})y_2] - B_1^{-1} k_n \tag{2.27}$$

Again, u has the form of (2.2) with:

$$\hat{u} = -B_1^{-1}[(A_{11} + C_2^T A_{21})y_1 + (A_{12} + C_2^T A_{22})y_2] \tag{2.28}$$

$$\bar{u} = -B_1^{-1} k_n \tag{2.29}$$

2. Transformed Equations on the Sliding Surfaces

When the system is on the sliding surfaces, the control laws become, $u = \hat{u}$, and the sliding surfaces after solving for y_1 are:

$$y_1 = -C_2^T y_2 \tag{2.30}$$

Using (2.28) and (2.30) in (2.21), the first set of system equations for $\dot{y}_1$ on the sliding surfaces reduces to:

$$-C_2^T \dot{y}^2 = -C_2^T (A_{21} y_1 + A_{22} y_2) \tag{2.31}$$

This is nothing more than a linear combination of the second set of system equations (2.22) for $\dot{y}_2$! Thus, there are only (n-m) independent equations on the sliding surfaces which gives m poles at the origin as before. The (n-m) independent equations on the sliding surfaces are:

$$\dot{y}_2 = (A_{22} - A_{21} C_2^T) y_2 \tag{2.32}$$

3. LQR Method for Determination of the Sliding Surfaces

In [Ref. 6], Utkin discusses pole placement and LQR methods for determining C_2^T. Both methods have been utilized for the control of the SDV during this study. The theory of pole placement is well known and will not be restated here.

For the LQR method, it is desired to minimize the quadratic performance index:

$$I = \frac{1}{2}\int_{t_0}^{\infty} x^T Q x \, dt = \frac{1}{2}\int_{t_0}^{\infty} y^T (T Q T^T) y \, dt \tag{2.33}$$

where, $Q \geq 0$

$$Q^T = Q$$

$t_0 \equiv$ time sliding mode begins

Partition the transformed weighting matrix such that:

$$T Q T^T = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{2.34}$$

Using the partitioned matrix (2.34) in (2.33) yields:

$$I = \frac{1}{2}\int_{t_0}^{\infty} (y_1^T Q_{11} y_1 + y_2^T Q_{21} y_1 + y_1^T Q_{12} y_2 + y_2^T Q_{22} y_2) dt \tag{2.35}$$

or:

$$I = \frac{1}{2}\int_{t_0}^{\infty} (y_2^T Q^* y_2 + v^T Q_{11} v) dt \tag{2.36}$$

where, $Q^* \equiv Q_{22} - Q_{21} Q_{11}^{-1} Q_{12}$

$$A^* \equiv A_{22} - A_{21} Q_{11}^{-1} Q_{12}$$

$$v \equiv y_1 + Q_{11}^{-1} Q_{12} y_1$$

Equation (2.36) is now a recognizable form for applying the LQR technique. One more step remains; the new definitions need to be applied to the system which is on the sliding surfaces. After these substitutions (2.32) converts to:

$$\dot{y}_2 = A^* y_2 + A_{21} v \tag{2.37}$$

The alterations are complete and the problem is now to minimize (2.36) subject to (2.37). The Hamiltonian for this problem is:

$$H = p^T (A^* y_2 + A_{21} v) - \frac{1}{2}(y_2^T Q^* y_2 + v^T Q_{11} v) \tag{2.38}$$

and the algebraic Riccati equation is:

$$A^{*T} k + kA - kA_{21} Q_{11}^{-1} A_{21}^T k + Q^* = 0 \tag{2.39}$$

which results in the solution:

$$v = -Q_{11}^{-1} A_{21}^T k y_2 \tag{2.40}$$

Again the new definitions from (2.36) are applied but in reverse to return to the original matrix variables. This produces the desired result:

$$C_2^T = Q_{11}^{-1}(Q_{12} + A_{21}^T k) \tag{2.41}$$

4. Determination of the Control Laws

After determining C_2^T by pole placement or by the LQR method above, calculation of the control laws is trivial by substitution into (2.28) and (2.29).

D. CONTROL OF CHATTERING

Although sliding mode is a robust means of control in the presence of parameter variations and input disturbances, it has an inherent chattering problem. This is caused by the discontinuity of the nonlinear feedbacks, $\bar{u}$. As stated earlier, these are switching terms with their signs switching between plus and minus according to the location of the system with respect to the sliding surfaces, such as:

$$sign(\sigma_i) \equiv \begin{cases} +1 \; ; \; \sigma_i > 0 \\ -1 \; ; \; \sigma_i < 0 \end{cases} \tag{2.42}$$

The chattering problem is eliminated by introducing small boundary layers of thickness ϕ_i around each sliding surface. Due to the sliding surfaces being defined via a Liapunov function, the system is guaranteed to move into the boundary layers towards the sliding surfaces. Unfortunately, the dynamics of the system trajectory inside the boundary layers are only an approximation to the desired dynamics on the sliding surfaces. The advantage of the scheme is that the trajectory will not chatter close to the sliding surfaces.

The scheme is carried out by defining a saturation function in place of sign(σ_i):

$$satsgn(\sigma_i) \equiv \begin{cases} +1 \; ; \; \sigma_i > \phi_i \\ \dfrac{\sigma_i}{\phi_i} \; ; \; -\phi_i < \sigma_i < \phi_i \\ -1 \; ; \; \sigma_i < -\phi_i \end{cases} \tag{2.43}$$

Theoretically, each sliding surface can have a different boundary layer thickness; however, during this study the same boundary layer thickness was used for all sliding surfaces of a given system.

E. COMMENTS

Methods for designing a MIMO sliding mode controller have been discussed. To review Utkin's method, the sliding surfaces can be determined by pole placement or by the LQR technique. Both of these methods, in the most general case, require a transformation of the states. As previously discussed, in this study the $\boldsymbol{B}$ matrix was always in the correct form and transformation was never needed.

Although pole placement has the advantage of direct placement of the sliding surface poles, its disadvantage is a lack of direct control over the control methodology, i.e., with what hierarchy will the states be minimized. As will be seen later, this can be of great importance for AUV control and is the motivation for using the more complex LQR method when conditions warrant.

III. MIMO SLIDING MODE CONTROL OF THE AUV IN THE DIVE PLANE

A. NONLINEAR GOVERNING EQUATIONS IN THE DIVE PLANE

1. Modifications to the SDV Equations of Motion

As stated in Chapter I, the Mark IX SDV has been used in this thesis for theoretical study of AUV dynamics; however, the SDV has several dissimilarities with the newly constructed NPS AUV. First, the SDV has a third propeller mounted underneath for surface operations. Second, the SDV has no vertical or horizontal thrusters for hovering or low speed operations. Finally, the SDV has stern rudders only. [Ref. 7]

The SDV model from [Ref. 7] was modified to approximate the AUV geometry as closely as possible. First, in the equations of motion the terms for the "extra" third propeller were dropped. Second, for diving operations vertical tunnel thrusters were added to the model. Finally, a set of bow rudders were added to the model for MIMO steering control. Steering will be discussed in Chapter IV.

a. Tunnel Thrusters

Several decisions had to be made in adding tunnel thrusters to the SDV model. Commensurate with the size of the SDV, it was decided that the two vertical tunnel thrusters would have a maximum thrust of 5 lbf each, and they would be located symmetrically on each side of the origin of the body axis coordinate system a longitudinal distance of 6.7 ft. It was also assumed that current would be used to

control the thrusters where +20 amperes would correspond to the maximum thrust of 5 lbf in the positive heave direction (downward). The relationship between thruster current and the resultant thrust was assumed to be linear [Ref. 8].

The thrusters were assumed to be infinitely responsive, i.e., zero response time. This is a realistic assumption for small thrusters.

The effect of vehicle surge velocity on resultant thrust is unknown for very small tunnel thrusters and is the topic of future studies at NPS. From studies of thrusters in general, it is known that thrust will decrease with increasing surge velocity [Ref. 9]. It was assumed during this study that the relationship was linear with a thrust degradation of 20% at maximum surge velocity.

Additionally, the effect of the thrusters on the surge velocity is also unknown and was unmodeled in this study.

2. Identification of Symbols

The six degree of freedom equations of motion for an underwater vehicle are usually described using a body fixed coordinate system and an inertial reference frame. Vehicle position is expressed in Cartesian coordinates x,y, and z. Orientation of the coordinate system is expressed in Euler angles ϕ, θ, and ψ. Hydrodynamic force components along the body axes are expressed as X, Y, and Z. Hydrodynamic moment components along the body axes are expressed as K, M, and N. Figure 8 of [Ref. 4] shows the positive directions of forces, moments, motions, and control surface deflections. The definitions of the states, controls, and other variables are listed in Tables 1, 2, and 3, respectively.

Table 1. DEFINITION OF AUV STATES

STATE	DEFINITION	UNITS
u	surge velocity	ft/s
v	sway velocity	ft/s
w	heave velocity	ft/s
p	roll rate	rad/s
q	pitch rate	rad/s
r	yaw rate	rad/s
ϕ	roll angle	radians
θ	pitch angle	radians
ψ	yaw angle	radians

Table 2. DEFINITION OF AUV CONTROLS

CONTROL	DEFINITION	UNITS
δ_{br}	bow rudder angle	radians
δ_{sr}	stern rudder angle	radians
δ_{bp}	bow plane angle	radians
δ_{sp}	stern plane angle	radians
I_{bv}	vert bow thruster current	amperes
I_{sv}	vert stern thruster current	amperes
N	propeller speed	rpm

Table 3. DEFINITION OF ADDITIONAL VARIABLES

VARIABLE	DEFINITION	UNITS
W	vehicle weight	lbf
B	buoyant force on vehicle	lbf
x	distance from the origin	ft
$b(x)$	vehicle beam at position x	ft
$h(x)$	vehicle height at position x	ft
x_g, y_g, z_g	location of CG	ft
x_B, y_B, z_B	location of CB	ft
$U_{cf}(x)$	cross flow velocity at x	ft/s
L	vehicle length	ft
ρ	density of medium	slugs/ft^3
υ	kinematic viscos. of medium	ft^2/s
m	vehicle mass	slugs
I_y	moment of inertia	slug-ft^2

3. Assumptions

The following assumptions have also been made for the diving model:

- Vehicle motion is confined to the vertical plane:

$$(p, \dot{p}, v, \dot{v}, \phi, \psi, r, \dot{r}, \delta_{br}, \delta_{sr} = 0)$$

- The AUV is neutrally buoyant:

$$(W = B)$$

- The CB and CG are on the vehicle's centerline:

$$(y_g, y_B = 0)$$

- The AUV is restricted to the speed range:

$$(-6.0\ ft/s \le u \le +6.0\ ft/s)$$
$$(-500\ rpm \le N \le +500\ rpm)$$

- The planes have a restricted range of motion:

$$|\delta_{bp}| \le 0.4\ radians$$
$$|\delta_{sp}| \le 0.4\ radians$$

- Surge velocity can be assumed constant in the heave and pitch equations:

$$(\dot{u} = 0)$$

These assumptions greatly simplify the governing nonlinear equations.

4. Resulting Equations

After all modifications and assumptions, the nonlinear governing equations from [Ref. 7] simplify to:

- Normal (Heave) equation of motion

$$\begin{aligned} m(\dot{w} - uq - x_g\dot{q} - z_g q^2) = \; & \frac{\rho}{2}L^4 Z'_{\dot{q}}\dot{q} \\ & + \frac{\rho}{2}L^3(Z'_{\dot{w}}\dot{w} + Z'_{q}uq) \\ & + \frac{\rho}{2}L^2[Z'_{w}uw + u^2(Z'_{\delta_{bp}}\delta_{bp} + Z'_{\delta_{sp}}\delta_{sp})] \\ & - \frac{\rho}{2}\int_{stern}^{bow}\frac{C_{Dz}\,b(x)(w - xq)^3}{U_{cf}(x)}dx \\ & + (0.25 - 0.008\bar{3}u)I_{bv} \\ & + (0.25 - 0.008\bar{3}u)I_{sv} \end{aligned} \tag{3.1}$$

- Pitch equation of motion

$$
\begin{aligned}
I_y\dot{q} - m[x_g(\dot{w} - uq) - z_g wq] = \\
&\frac{\rho}{2}L^5 M'_q \dot{q} \\
&+ \frac{\rho}{2}L^4 (M'_w \dot{w} + M'_q uq) \\
&+ \frac{\rho}{2}L^3 [M'_w uw + u^2(M'_{\delta_{bp}}\delta_{bp} + M'_{\delta_{sp}}\delta_{sp})] \\
&+ \frac{\rho}{2}\int_{\text{stern}}^{\text{bow}} \frac{C_{Dz}\, b(x)(w - xq)^3}{U_{cf}(x)} x\, dx \\
&- (x_g W - x_B B)\cos\theta \\
&- (z_g W - z_B B)\sin\theta \\
&- 6.7(0.25 - 0.008\bar{3}u) I_{bv} \\
&+ 6.7(0.25 - 0.008\bar{3}u) I_{sv}
\end{aligned}
\tag{3.2}
$$

- Surge equation of motion

$$
\begin{aligned}
m(\dot{u} + wq - x_g q^2 + z_g \dot{q}) = \\
&\frac{\rho}{2}L^4 X'_{qq} qq \\
&+ \frac{\rho}{2}L^3 [X'_{\dot{u}}\dot{u} + X'_{wq} wq + uq(X'_{q\delta_{bp}}\delta_{bp} + X'_{q\delta_{sp}}\delta_{sp})] \\
&+ \frac{\rho}{2}L^2 [X'_{ww} w^2 + uw(X'_{w\delta_{bp}}\delta_{bp} + X'_{w\delta_{sp}}\delta_{sp})] \\
&+ \frac{\rho}{2}L^2 u^2 (X'_{\delta_{bp}\delta_{bp}}\delta_{bp}^2 + X'_{\delta_{sp}\delta_{sp}}\delta_{sp}^2) \\
&+ \frac{\rho}{2}L^2 u^2 X'_{prop}
\end{aligned}
\tag{3.3}
$$

- Kinematic equations

$$\dot{\theta} = q \tag{3.4}$$

$$\dot{z} = w - u\sin\theta \tag{3.5}$$

- Propulsion equations

$$X_{prop} \equiv C_{D0}[\eta\,|\eta| - 1] \tag{3.6}$$

$$C_{D0} \equiv 0.00385 + 1.296\times10^{-17}(R_e - 1.2\times10^7)^2 \tag{3.7}$$

$$R_e \equiv \frac{uL}{\nu}\ ;\ \textit{Reynold's number} \tag{3.8}$$

$$\eta \equiv \frac{u_d}{u} \tag{3.9}$$

$$u_d \equiv 0.012\,N\ ;\ \textit{desired surge velocity} \tag{3.10}$$

- Cross flow velocity equation

$$U_{cf}(x) = |(w - xq)| \tag{3.11}$$

B. CONTROL METHODOLOGY

Previous theses at NPS have attempted to control the AUV only at high speeds in the dive plane, e.g., [Ref. 4]. Problems arise at low speeds with only planes as control inputs. In fact, depth of the neutrally buoyant AUV cannot be controlled at speeds less than approximately 1.0 ft/s because the hydrodynamic forces on the planes become negligible. This control problem can be overcome by the addition of thrusters; however, this poses new control problems. With two sets of planes and two thrusters, the system now has four control inputs! The problem arises as to how

these inputs should be utilized for efficient vehicle control. Figure 1 compares the hydrodynamic forces produced by the planes and thrusters versus surge velocity. Obviously, at very low speeds the planes are of little use. At high speeds, the thrust produced by the tunnel thrusters is negligible in comparison to the planes.

For efficient control, it was decided to use different control laws at different surge velocities. The surge velocity u was partitioned into three distinct regions:

- Hover region: $0.00 \leq u \leq 0.75$ ft/s
- Transition region: $0.75 < u \leq 1.75$ ft/s
- Cruise region: $1.75 < u \leq 6.00$ ft/s

In the hover region, only thrusters are used for vehicle control. In the transition region, thrusters and planes are used. Finally, in the cruise region planes are utilized. Thus, there are unique control laws and sliding surfaces in each region.

The simplified nonlinear governing equations from the previous section are linearized at the midpoint of each speed region, $u = 0.375, 1.25, 3.875$ ft/s, to obtain the linearized state space representation of (2.1) for controller design. Due to the complexity of the heave (3.1) and pitch (3.2) equations, a Fortran program was written to assist in their solution. The program is named SDVDIVLIN_I.FOR (see Appendix A). The program solves the heave and pitch equations for $\dot{w}$ and $\dot{q}$. Program output is expressed as two highly nonlinear equations:

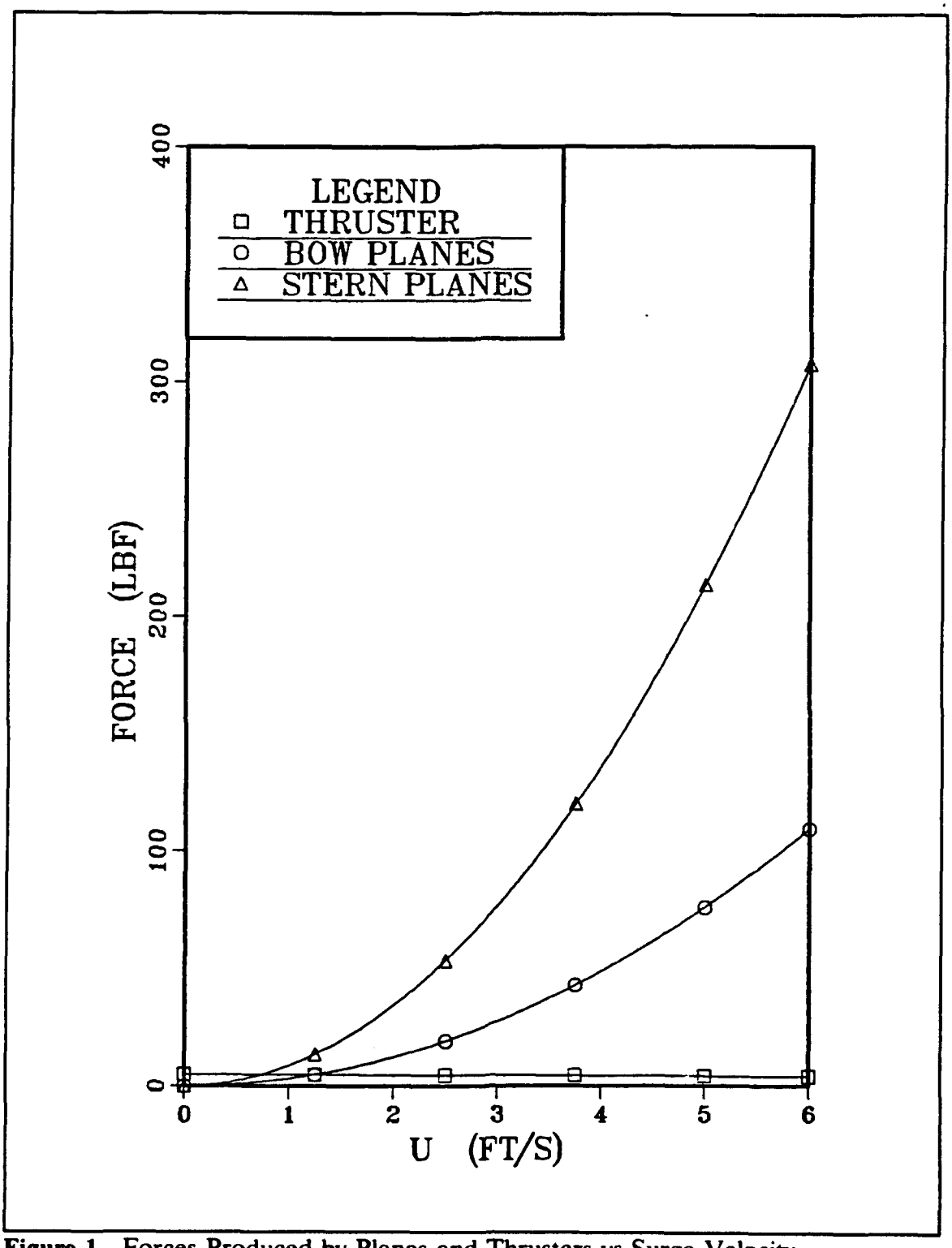


Figure 1. Forces Produced by Planes and Thrusters vs Surge Velocity

$$\frac{d}{dt}x(t) = M^{-1}f(x(t), u(t)) \tag{3.12}$$

where, $M \equiv$ "mass" matrix

The heave and pitch equations are then linearized, as is kinematic equation (3.5). With the inclusion of the linear kinematic definition (3.4), the resultant system has the form of (2.1):

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = [A]\begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} + [B]\begin{bmatrix} \delta_{bp} \\ \delta_{sp} \\ I_{bv} \\ I_{sv} \end{bmatrix} \tag{3.13}$$

The sliding surfaces and control laws of the AUV are then based on this linear system.

C. CONTROL OF THE AUV IN THE HOVER REGION

1. State Space Equations

To reiterate, in the hover region, $0.00 \leq u \leq 0.75$ ft/s, the planes are not used and the nonlinear system is linearized at $u = 0.375$ ft/s to determine the sliding surfaces and control laws. The resulting linear state space representation is:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -0.0227 & -0.0505 & 0.0251 & 0.0 \\ 0.0056 & -0.0627 & -0.0668 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & -0.3750 & 0.0 \end{bmatrix}\begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} 0.00017 & 0.00014 \\ -0.00005 & 0.00004 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}\begin{bmatrix} I_{bv} \\ I_{sv} \end{bmatrix} \tag{3.14}$$

Note the $\boldsymbol{B}$ matrix has the required form of (2.16) and the transformation matrix $\boldsymbol{T}$ discussed in Chapter II is not needed, or $\boldsymbol{T}$ can be thought of as the identity matrix.

Using Utkin's method of MIMO sliding mode controller design, (3.14) can be expressed in the form of (2.21) and (2.22) with:

$$\boldsymbol{A}_{11} = \begin{bmatrix} -0.0227 & -0.0505 \\ 0.0056 & -0.0627 \end{bmatrix} \qquad \boldsymbol{A}_{12} = \begin{bmatrix} 0.0251 & 0.0 \\ -0.0668 & 0.0 \end{bmatrix}$$

$$\boldsymbol{A}_{21} = \begin{bmatrix} 0.0 & 1.0 \\ 1.0 & 0.0 \end{bmatrix} \qquad \boldsymbol{A}_{22} = \begin{bmatrix} 0.0 & 0.0 \\ -0.3750 & 0.0 \end{bmatrix}$$

$$\boldsymbol{B}_1 = \begin{bmatrix} 0.00017 & 0.00014 \\ -0.00005 & 0.00004 \end{bmatrix} \tag{3.15}$$

$$\boldsymbol{y}_1 = \begin{bmatrix} w & q \end{bmatrix}^{\mathrm{T}} \qquad \boldsymbol{y}_2 = \begin{bmatrix} \theta & z \end{bmatrix}^{\mathrm{T}} \qquad \boldsymbol{u} = \begin{bmatrix} I_{\mathrm{bv}} & I_{\mathrm{sv}} \end{bmatrix}^{\mathrm{T}}$$

The MIMO sliding mode controller will be based on this linear system.

2. Controller Design by Pole Placement

With pole placement, the first step is to place the poles of the system on the sliding surfaces to determine $\boldsymbol{C}_2^{\mathrm{T}}$:

$$POLEPLACE\left\{\boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{C}_2^{\mathrm{T}}\right\} \tag{3.16}$$

The reader is reminded that this method requires $\boldsymbol{C}_1^{\mathrm{T}}$ to be set equal to the 2×2 identity matrix as discussed in Chapter II.

A problem arises because C_2^T is a 2×2 matrix with four unknowns, while (3.16) turns out to be a simple quadratic. Thus, there are two equations with four unknowns. This is a major disadvantage of using pole placement for MIMO systems; there are more unknowns than equations which precludes a straight forward use of Matrix-x to easily calculate the sliding surfaces. The problem can be overcome by additional constraints on the eigenvectors or by assuming relationships between the elements of C_2^T. Here we arbitrarily assumed the two relationships:

$$\begin{aligned} C_2(1,1) &= 0.5\,C_2(1,2) \\ C_2(2,1) &= 0.5\,C_2(2,2) \end{aligned} \tag{3.17}$$

Additionally, we pick the two poles at -0.50 and -0.51. C_2^T can now be determined:

$$C_2^T = \begin{bmatrix} 0.6750 & -0.3400 \\ 1.3500 & -0.6800 \end{bmatrix} \tag{3.18}$$

With (3.18), the two sliding surfaces are:

$$\begin{aligned} \sigma_1 &= 1.0\,w + 0.6750\,\theta - 0.3400\,z \\ \sigma_2 &= 1.0\,q + 1.3500\,\theta - 0.6800\,z \end{aligned} \tag{3.19}$$

Here we define the states to be interpreted as errors between their actual and desired values. With this in mind, z, can be written as, $z - z_d$.

Now that the sliding surfaces have been calculated, the control laws are determined from (2.27):

$$
\begin{aligned}
I_{bv} &= -(0.5420\times10^4)\,w + (1.0557\times10^4)\,q + (0.1354\times10^4)\,\theta \\
&\quad - (0.3097\times10^4)\,\eta^2\,satsgn(\sigma_1) + (0.9703\times10^4)\,\eta^2\,satsgn(\sigma_2) \\
I_{sv} &= (0.9463\times10^4)\,w - (1.7829\times10^4)\,q - (0.2819\times10^4)\,\theta \\
&\quad - (0.3464\times10^4)\,\eta^2\,satsgn(\sigma_1) - (1.2169\times10^4)\,\eta^2\,satsgn(\sigma_2)
\end{aligned}
\tag{3.20}
$$

The reader should note that the same η^2 has been used in front of both *satsgn* functions for simplicity, although this is not required.

3. Computer Simulation with Pole Placement Controller

Fortran program SDVDIVPPTTPSF_POLE.FOR (see Appendix A) was used to simulate the AUV in the dive plane for all three regions - hover, transition, and cruise - using controllers designed by pole placement. The program assumes perfect state feedback. Drag on the AUV is calculated in the program by the trapezoidal rule. The surge velocity of the AUV is controlled in the program by the speed controller developed by Lienard [Ref. 3]. The simulation time step is $\Delta t = 0.01$ s.

Fortran program DPLOT2.FOR (see Appendix A) utilizes the raw data generated by SDVDIVPPTTPSF_POLE.FOR and plots several graphs to visualize the simulation. DPLOT2.FOR calls plotting subroutines from the software program DISSPLA (Copyright 1988 by Computer Associates International, Inc.). Figures 2 and 3 were generated by DPLOT2.FOR and show the results of Run 1 at $u = 0.25$ ft/s with $\phi = 2.0$ and $\eta^2 = 0.2$.

Figures 2 and 3 clearly show the pole placement controller doesn't work. The bow thruster forces the bow down while the stern thruster lifts the stern. This

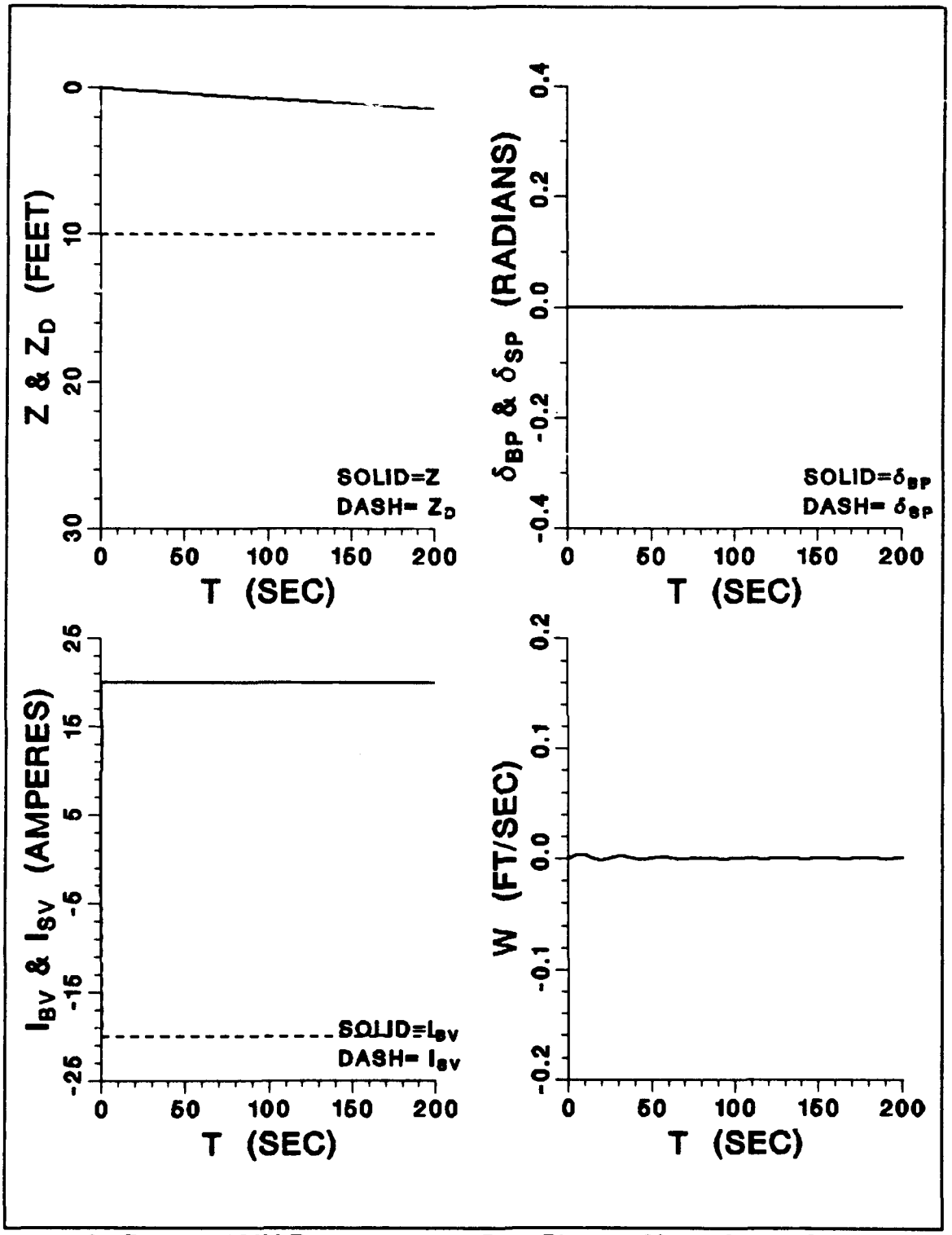


Figure 2. Run 1 - AUV Response in the Dive Plane at Hover Speed, Pole Placement (u = 0.25 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

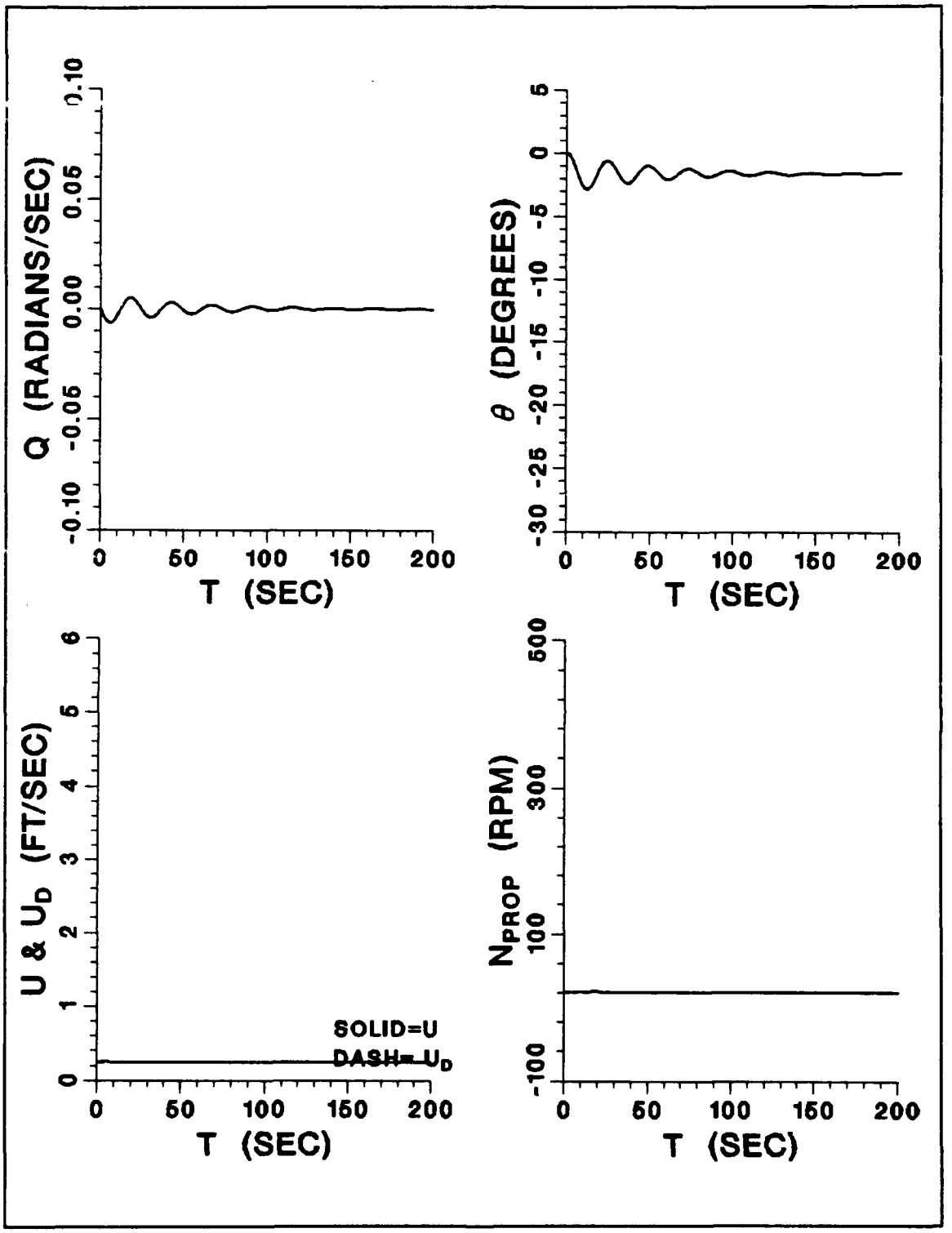


Figure 3. Run 1 - (continued)

pitching scheme would work well at higher speeds when the AUV has surge velocity to reach depth; however, both thrusters need to work in the same direction in the hover region.

What went wrong? At speeds in the hover region, we need to minimize q and θ. This ensures the vehicle remains flat and the thrusters work in the same direction to force the vehicle down to ordered depth; however, pole placement doesn't allow the designer to choose which states to minimize. There are an infinite number of solutions for C_2^T and not all of them may be effective. The two relationships (3.17) we assumed to solve for C_2^T obviously didn't lead to the desired control action at hover speeds; although, they may work at higher speeds with planes. A designer could assume numerous relationships/values and conceivably never stumble upon the correct combination.

It's obvious that simple pole placement, as formulated here, did not work in this situation. The next step is to use the LQR technique as discussed in Chapter II.

4. Controller Design by LQR Technique

A Matrix-x program was written to assist in the calculation of the sliding surfaces and control laws via the LQR technique (see Appendix B). This program was modified for all subsequent LQR controller designs. The designer interactively enters the diagonal values of the minimization matrix Q. As discussed in the previous section, minimizing q and θ should yield sliding surfaces and control laws that result in the correct control action in the hovering region. To minimize these states, the

values $Q_{11} = 5$, $Q_{22} = 1000$, $Q_{33} = 500$, and $Q_{44} = 1$ were selected after some experimentation. The resulting sliding surfaces are:

$$\begin{aligned} \sigma_1 &= 1.0\,w - 0.1452\,\theta + 0.4471\,(z - z_d) \\ \sigma_2 &= 1.0\,q + 0.7074\,\theta - 0.0007\,(z - z_d) \end{aligned} \tag{3.21}$$

and control laws:

$$\begin{aligned} I_{bv} &= -(1.2666\times10^3)\,w + (6.8616\times10^3)\,q - (0.2037\times10^3)\,\theta \\ &\quad - (0.3097\times10^4)\,\eta^2\,satsgn(\sigma_1) + (0.9703\times10^4)\,\eta^2\,satsgn(\sigma_2) \\ I_{sv} &= -(1.5299\times10^3)\,w - (7.1672\times10^3)\,q + (1.3030\times10^3)\,\theta \\ &\quad - (0.3464\times10^4)\,\eta^2\,satsgn(\sigma_1) - (1.2169\times10^4)\,\eta^2\,satsgn(\sigma_2) \end{aligned} \tag{3.22}$$

In addition to the two zero poles, this controller has poles of -0.4477 and -0.7068 on the sliding surfaces.

5. Computer Simulation with LQR Controller

Fortran program SDVDIVPPTTPSF_LQR.FOR (see Appendix A) was used to simulate the AUV in the dive plane for all three regions - hover transition, and cruise - using controllers designed by the LQR technique. The program is a duplicate of the pole placement simulation program with the only difference being the controllers. Fortran program DPLOT2.FOR was again used to plot graphs of the simulation. Figures 4, 5, 6, and 7 were generated by DPLOT2.FOR and demonstrate the results at two different speeds in the hover region. Figures 4 and 5 are of Run 2 at $u = 0.00$ ft/s, and Figures 6 and 7 are of Run 3 at $u = 0.55$ ft/s. Both runs were executed with values of $\phi = 2.0$ and $\eta^2 = 0.2$.

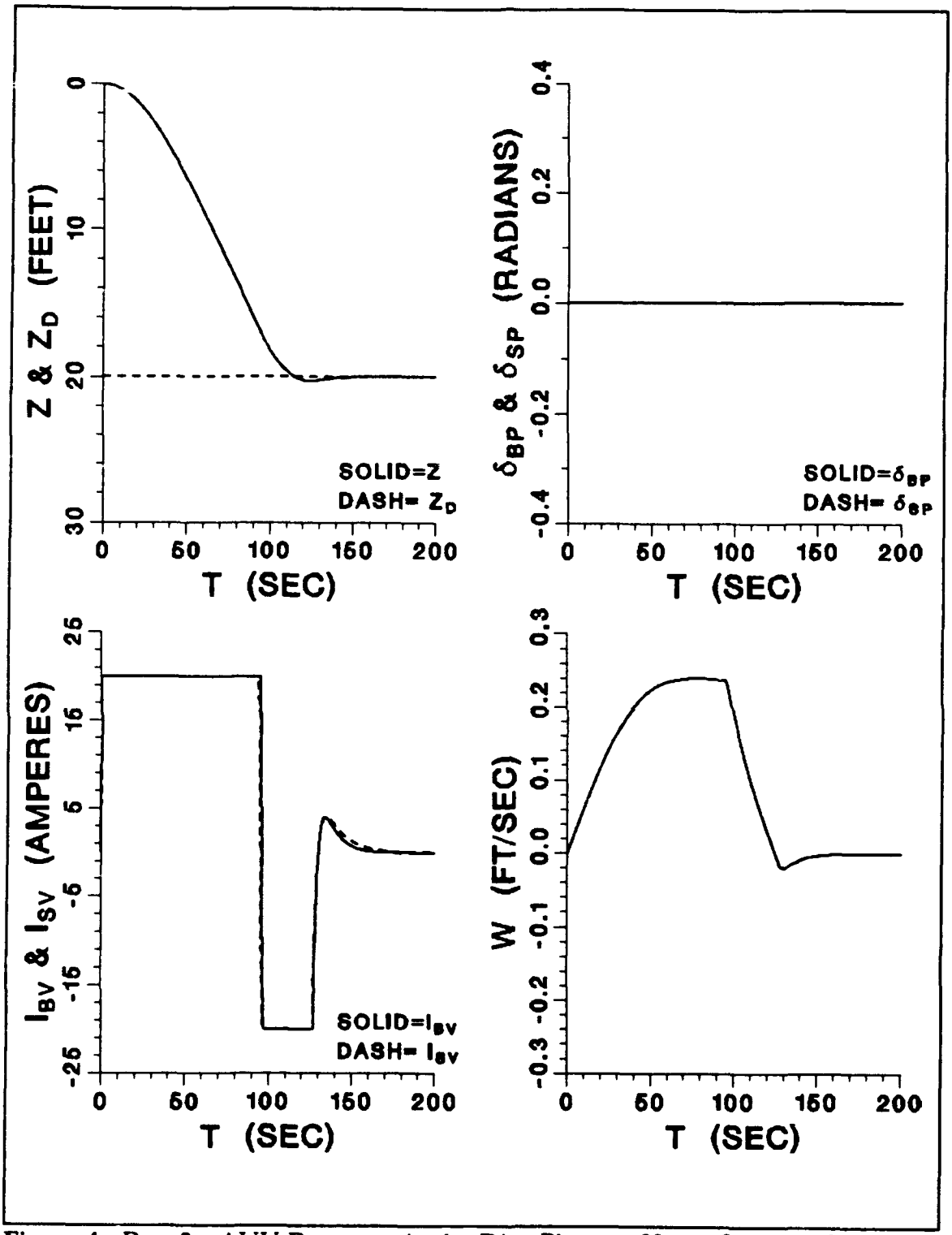


Figure 4. Run 2 - AUV Response in the Dive Plane at Hover Speed, LQR (u = 0.00 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

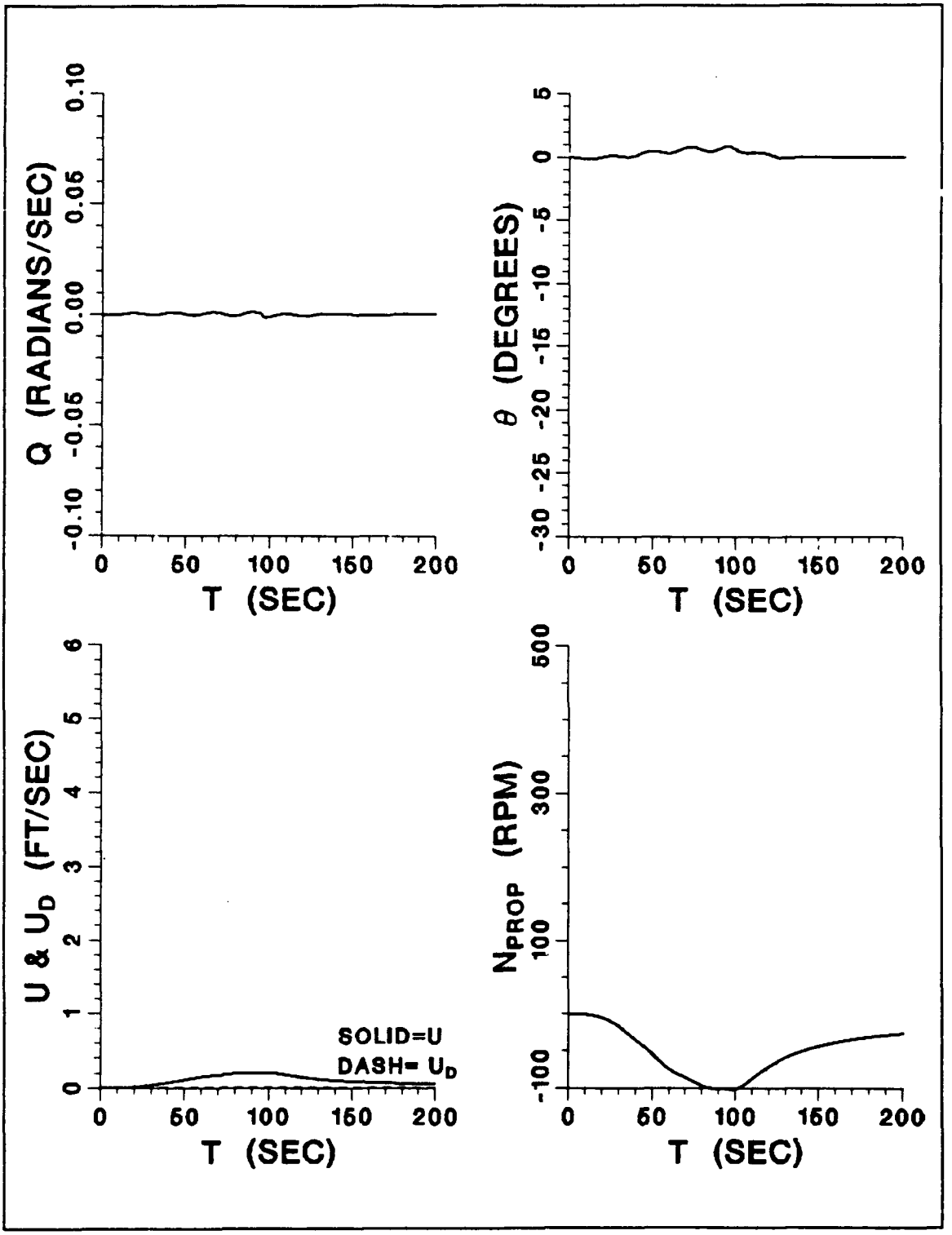


Figure 5. Run 2 - (continued)

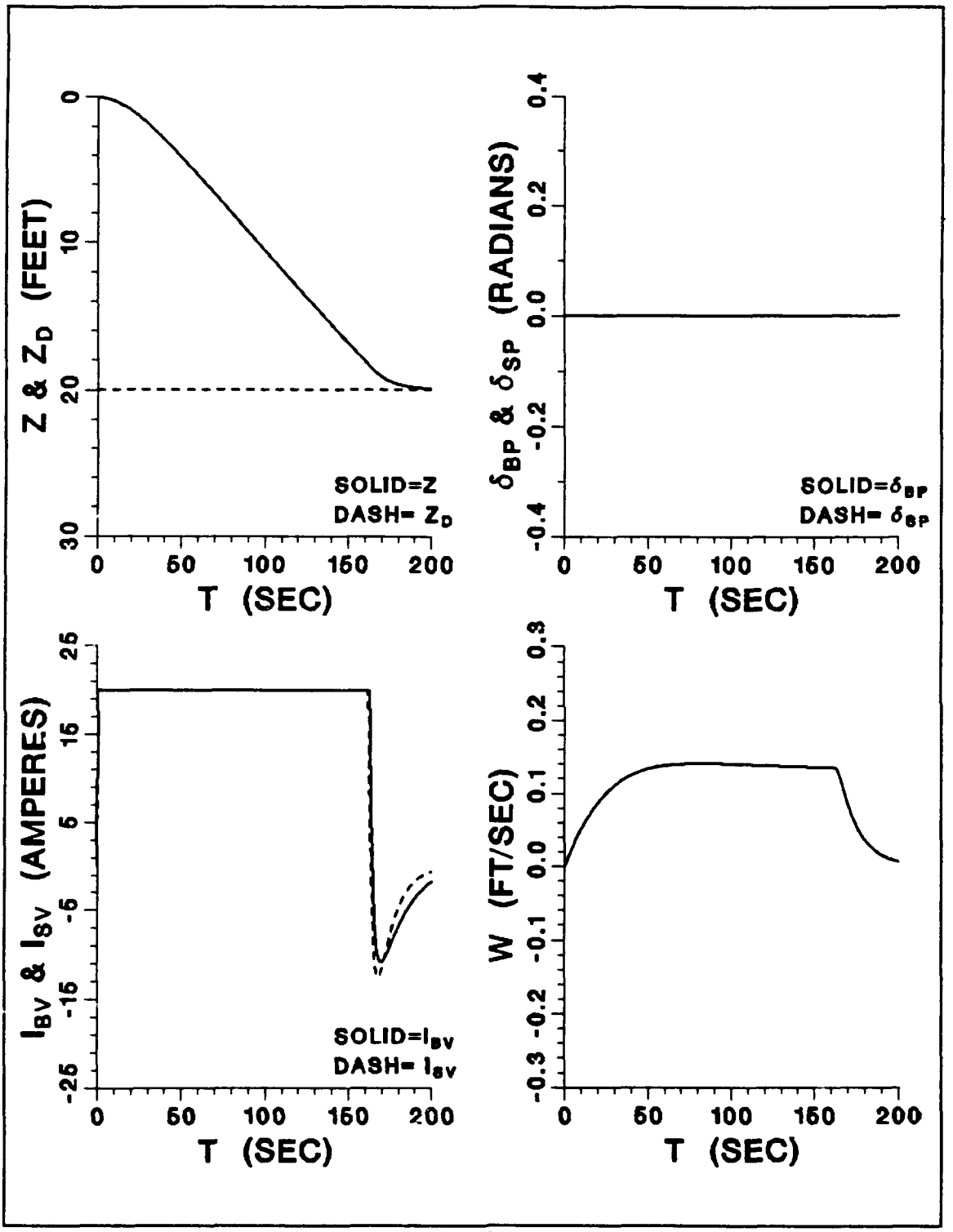


Figure 6. Run 3 - AUV Response in the Dive Plane at Hover Speed, LQR (u = 0.55 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

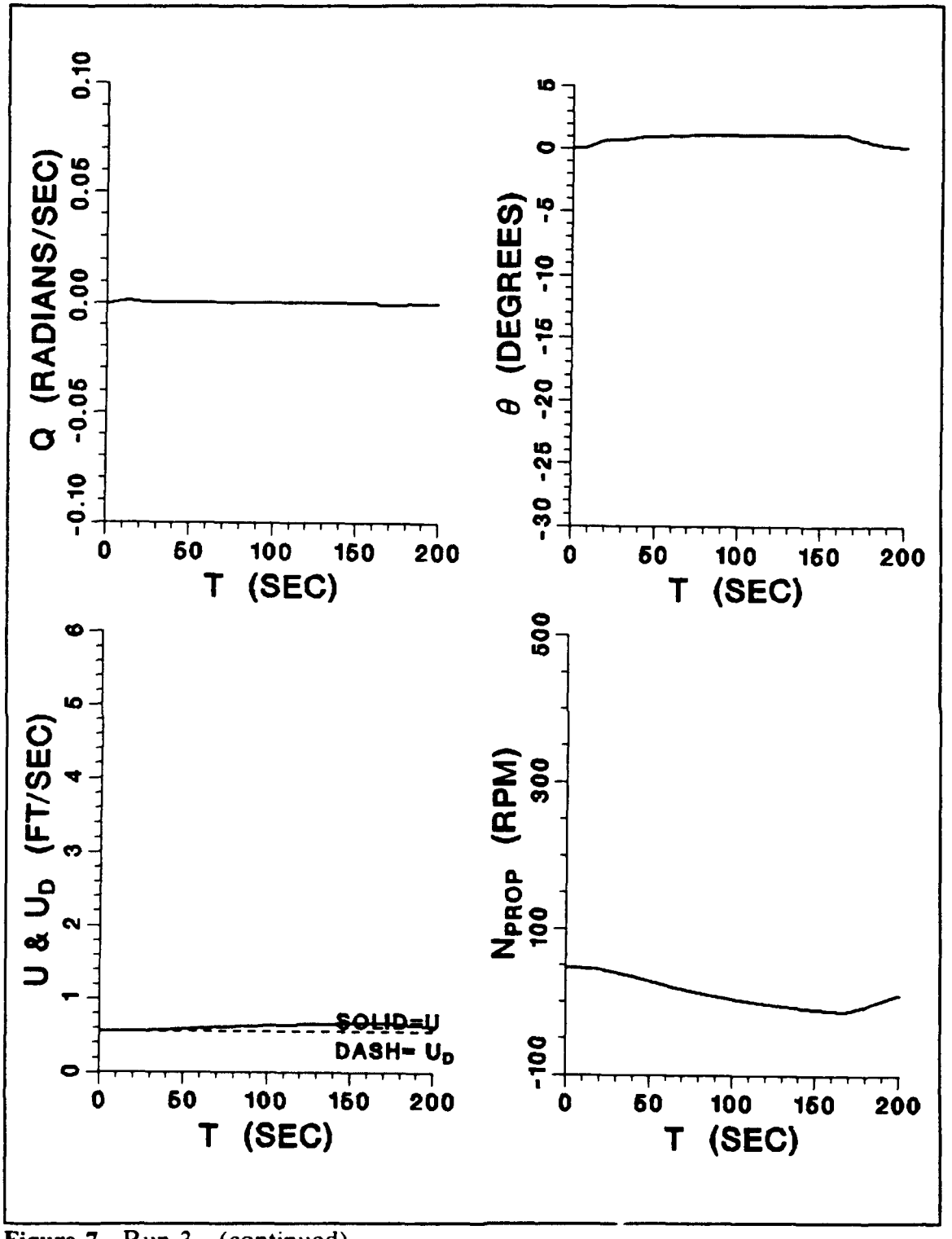


Figure 7. Run 3 - (continued)

Figures 4 to 7 conclusively show the LQR controller works satisfactorily. The graphs of q and θ reveal the vehicle has a minimal amount of pitch angle as it is thrusted downwards. The AUV expeditiously reaches its ordered depth with a minor amount of overshoot by using heave velocity w created by the thrusters working together.

Note the simulations show the AUV's surge velocity u increasing when the thrusters are energized due to the unmodeled effect of the thrusters on the surge velocity. Intuitively, this should not happen with the real vehicle.

D. CONTROL OF THE AUV IN THE CRUISE REGION

1. State Space Equations

To review, in the cruise region, $1.75 < u \leq 6.00$ ft/s, the thrusters are not used and the nonlinear system is linearized at $u = 3.875$ ft/s to determine the sliding surfaces and control laws. The resulting linear state space representation is:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -0.2346 & -0.5222 & 0.0251 & 0.0 \\ 0.0581 & -0.6483 & -0.0668 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & -3.8750 & 0.0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} -0.0765 & -0.1661 \\ 0.0161 & -0.0850 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \delta_{bp} \\ \delta_{sp} \end{bmatrix} \quad (3.23)$$

Again, the B matrix does not need to be transformed.

2. Controller Design by Pole Placement

It will be interesting to investigate if an effective controller can be designed by pole placement at cruise speeds. The sliding surfaces and control laws are

designed via pole placement identically with the procedure in the hover region. The same poles (-0.50, -0.51) are utilized, and the same two relationships (3.17) are assumed. The resultant sliding surfaces are:

$$\begin{aligned} \sigma_1 &= 1.0\,w + 0.5215\,\theta - 0.0329\,(z - z_d) \\ \sigma_2 &= 1.0\,q + 1.0429\,\theta - 0.0658\,(z - z_d) \end{aligned} \tag{3.24}$$

which yields the following control laws:

$$\begin{aligned} \delta_{bp} &= -2.3407\,w - 7.1582\,q - 1.9971\,\theta \\ &\quad + 9.2671\,\eta^2\,satsgn(\sigma_1) - 18.1220\,\eta^2\,satsgn(\sigma_2) \\ \delta_{sp} &= -0.5329\,w + 3.2909\,q + 1.8379\,\theta \\ &\quad + 1.7528\,\eta^2\,satsgn(\sigma_1) + 8.3426\,\eta^2\,satsgn(\sigma_2) \end{aligned} \tag{3.25}$$

3. Computer Simulation with Pole Placement Controller

Figures 8 and 9 were generated by SDVDIVPPTTPSF_POLE.FOR and DPLOT2.FOR and conclusively prove the pole placement controller works well in the cruise region. The figures exhibit Run 4 at u = 3.875 ft/s, ϕ = 1.0, and η^2 = 0.35. The AUV reaches ordered depth quickly with negligible overshoot.

Pole placement is obviously not a dependable procedure for designing MIMO sliding mode controllers and will not be used again during this thesis. The procedure we used to design a unsatisfactory controller in the hover region worked well when it was applied to design a controller in the cruise region, and even this may have been a matter of luck. The only advantage of pole placement is that it can guarantee negative real poles on the sliding surfaces, but as we saw in the hover

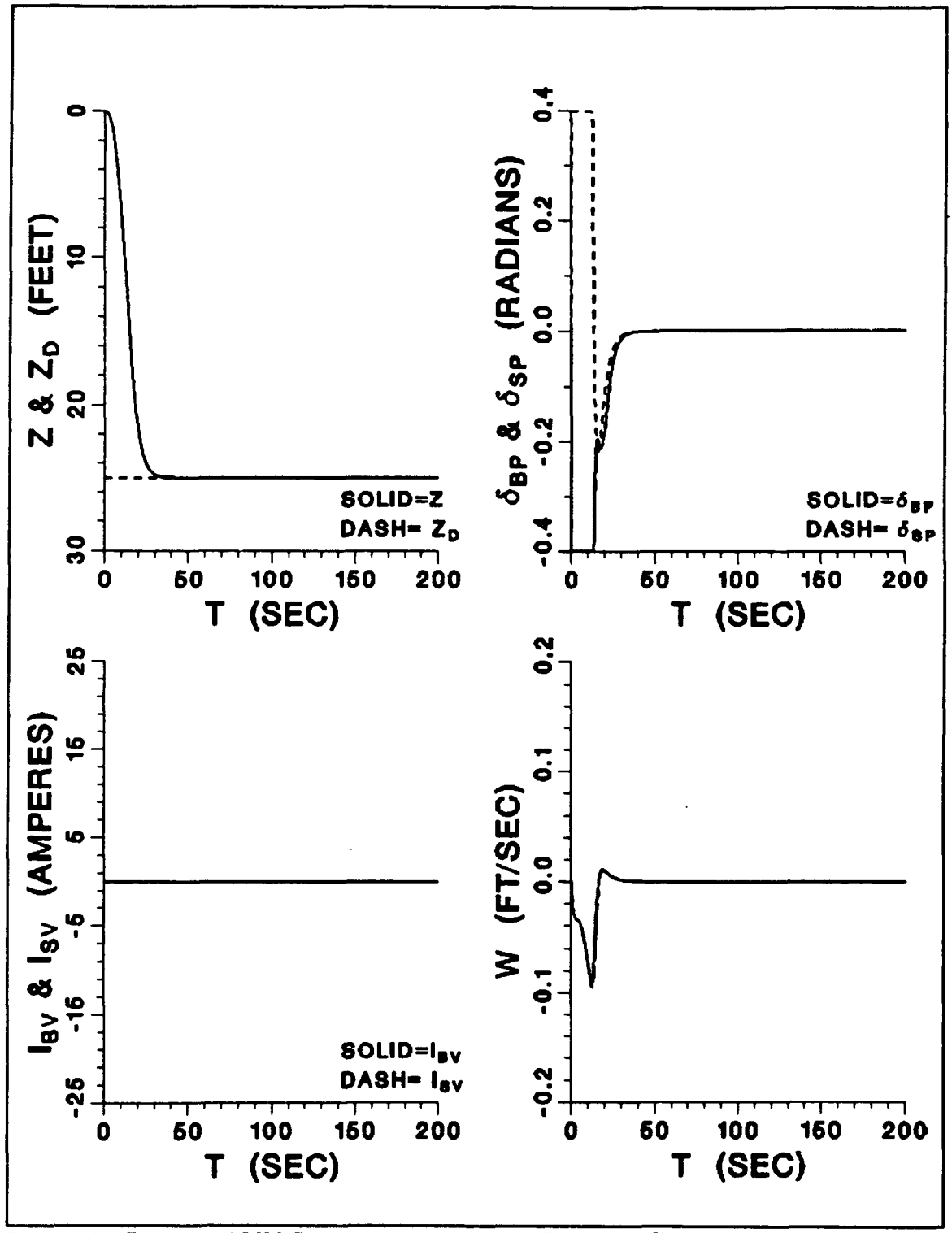


Figure 8. Run 4 - AUV Response in the Dive Plane at Cruise Speed, Pole Placement (u = 3.875 ft/s, ϕ = 1.0, η^2 = 0.35, Perfect State Feedback)

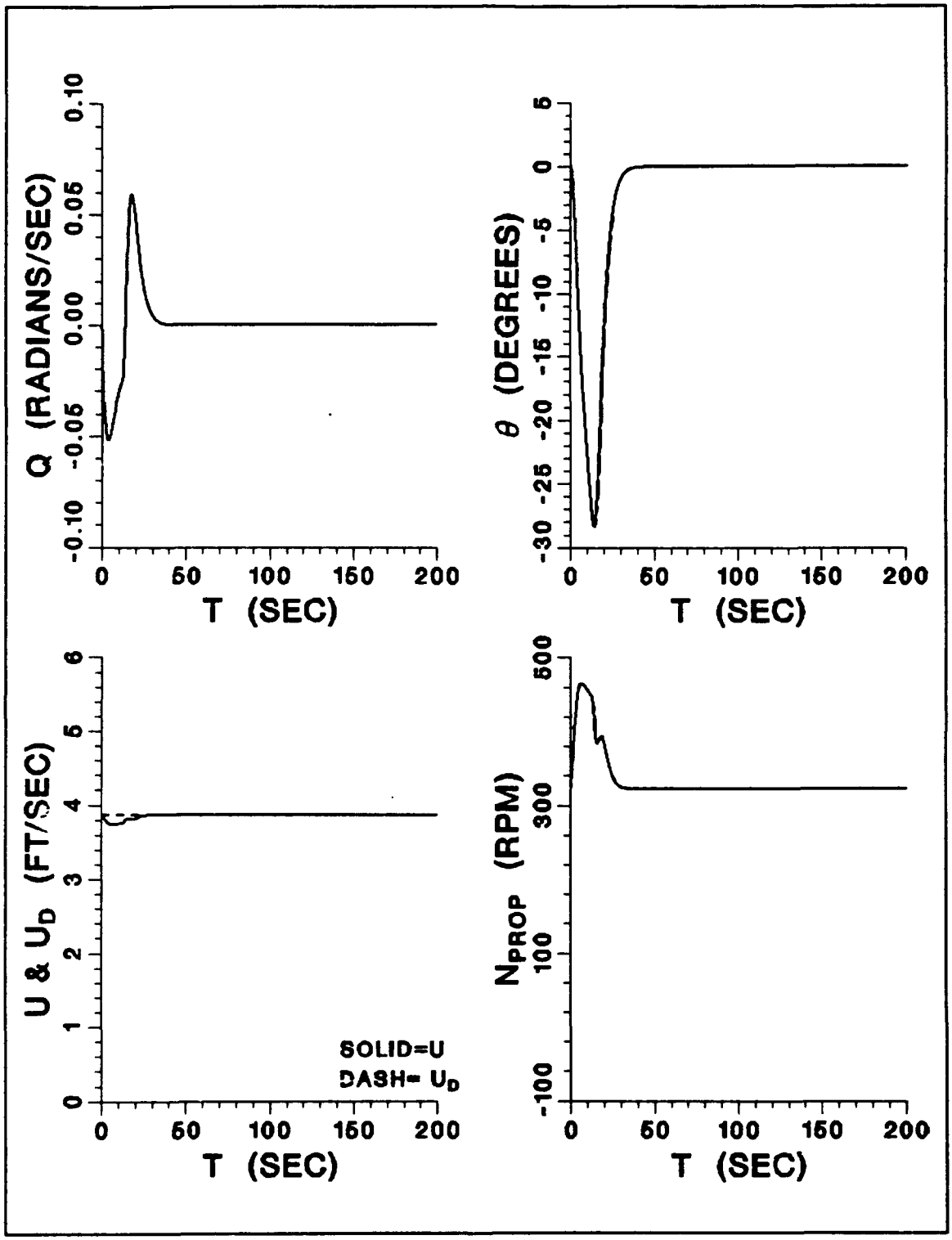


Figure 9. Run 4 - (continued)

region this may not be the most important consideration.

4. Controller Design by LQR Technique

In the cruise region, the designer is faced with a dilemma when using the LQR technique. Which states should be minimized to get the desired pitching control action? Extensive experimentation led to minimizing w, q, and θ as the best solution. The diagonal of the LQR minimization matrix Q was selected as $Q_{11} = 100$, $Q_{22} = 100$, $Q_{33} = 100$, and $Q_{44} = 1$. Matrix-x was again used to assist in the calculations. The resulting equations are:

$$\begin{aligned} \sigma_1 &= 1.0\,w - 0.0945\,\theta + 0.0328\,(z - z_d) \\ \sigma_2 &= 1.0\,q + 1.3127\,\theta - 0.0945\,(z - z_d) \end{aligned} \tag{3.26}$$

$$\begin{aligned} \delta_{bp} &= -1.2124\,w - 17.7554\,q - 6.3692\,\theta \\ &\quad + 9.2671\,\eta^2\,satsgn(\sigma_1) - 18.1220\,\eta^2\,satsgn(\sigma_2) \\ \delta_{sp} &= -0.6568\,w + 4.4622\,q + 2.3181\,\theta \\ &\quad + 1.7528\,\eta^2\,satsgn(\sigma_1) + 8.3426\,\eta^2\,satsgn(\sigma_2) \end{aligned} \tag{3.27}$$

The two nonzero poles of this controller are -0.4438 and -0.9017.

5. Computer Simulation with LQR Controller

Figures 10, 11, 12, and 13 show Runs 5 and 6 for the cruise LQR controller at 1.90 ft/s and 6.00 ft/s. The same values of ϕ, η^2, and z_d have been used as in Run 4 for the cruise pole placement controller. Again, the controller works well; the AUV reaches ordered depth quickly with negligible overshoot. As expected, the speed of response of the AUV increases with surge velocity due to the larger

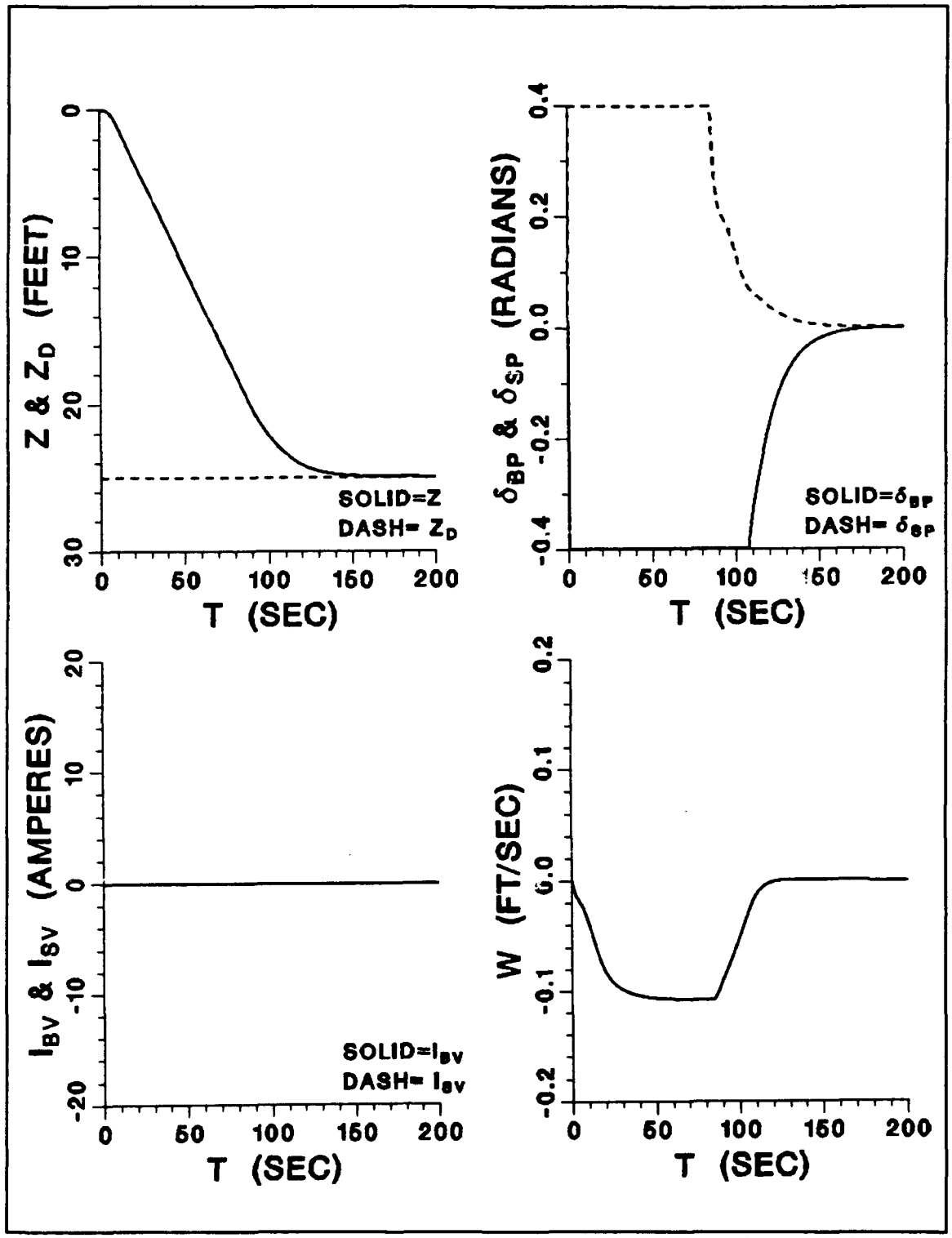


Figure 10. Run 5 - AUV Response in the Dive Plane at Cruise Speed, LQR (u = 1.90 ft/s, ϕ = 1.0, η^2 = 0.35, Perfect State Feedback)

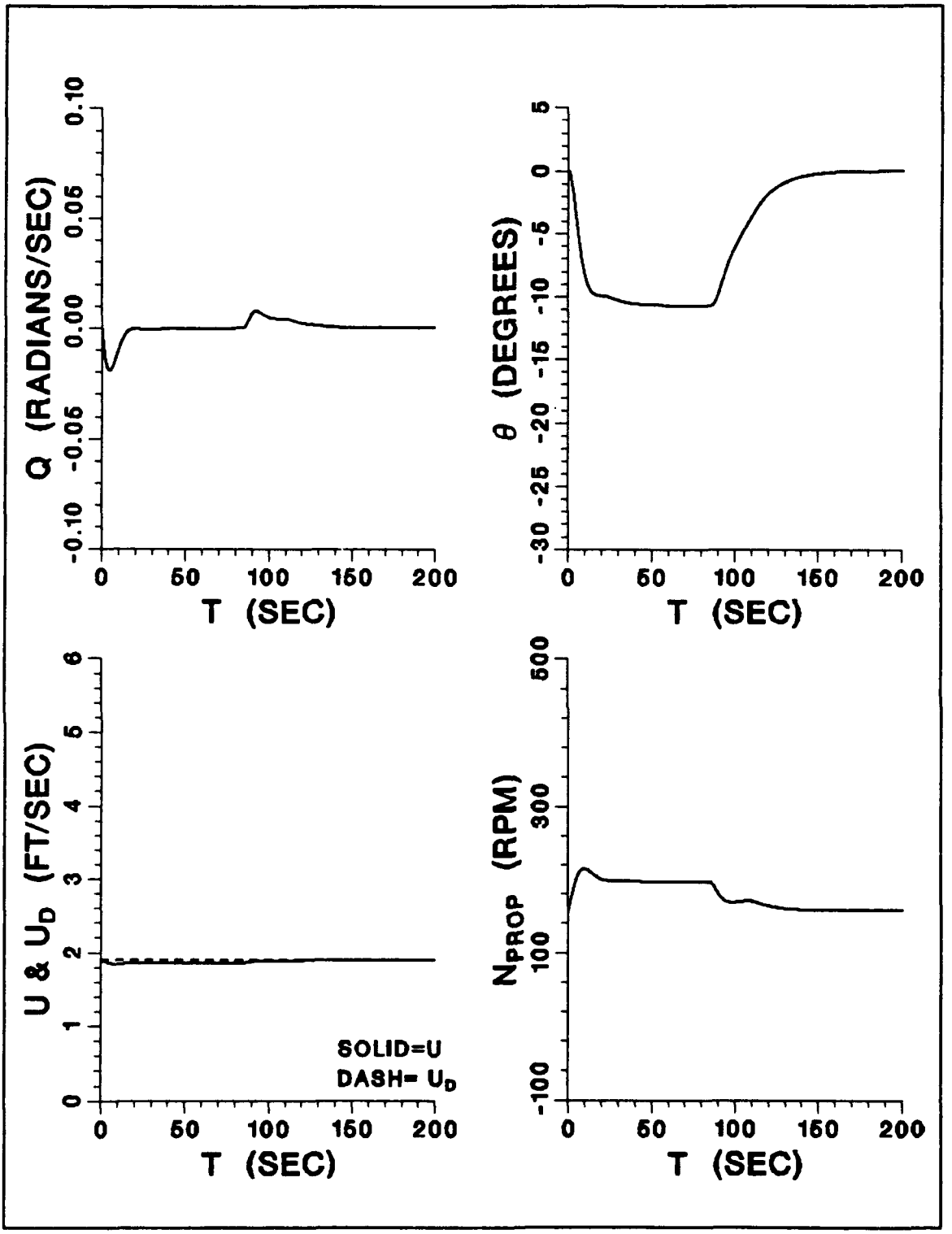


Figure 11. Run 5 - (continued)

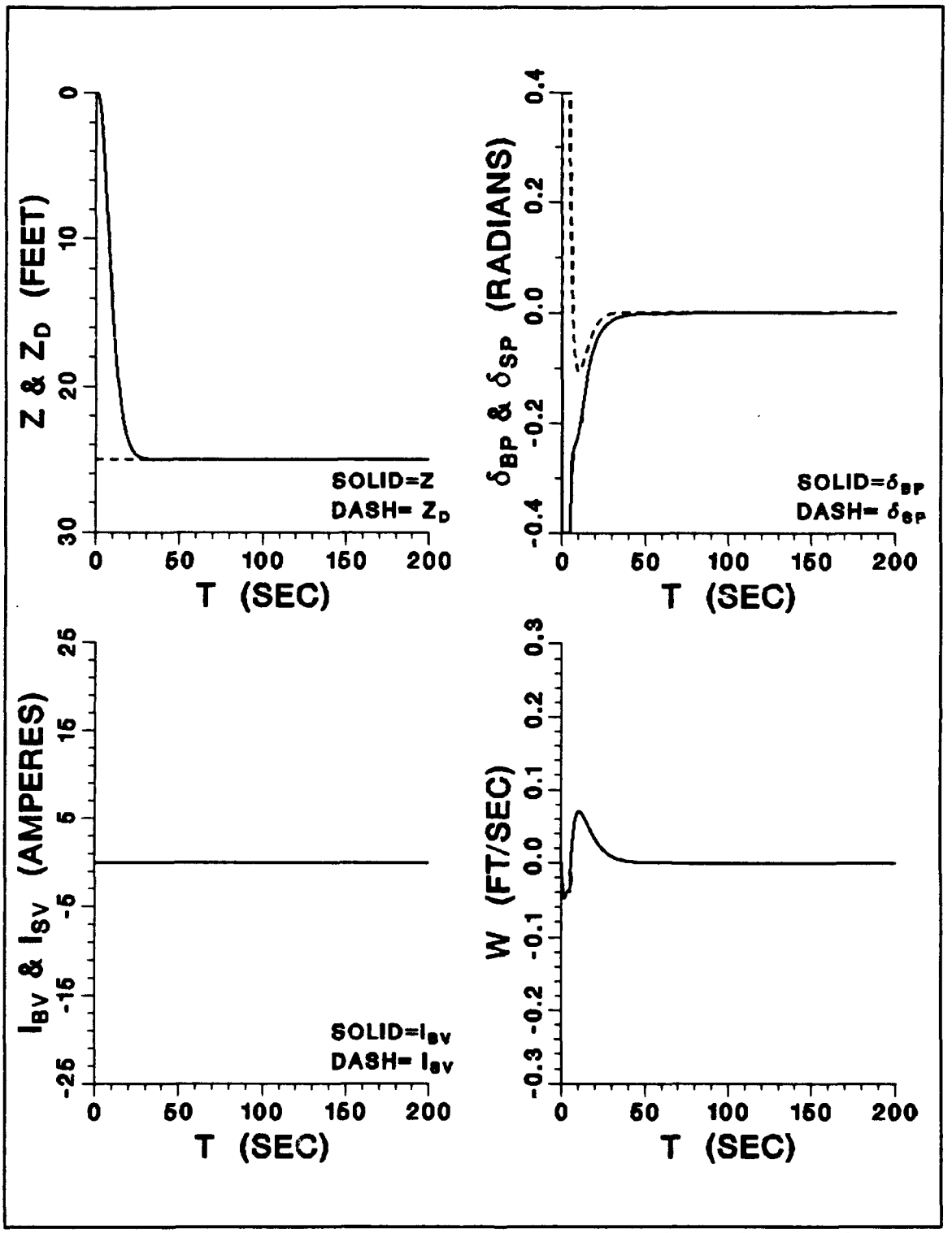


Figure 12. Run 6 - AUV Response in the Dive Plane at Cruise Speed, LQR (u = 6.00 ft/s, ϕ = 1.0, η^2 = 0.35, Perfect State Feedback)

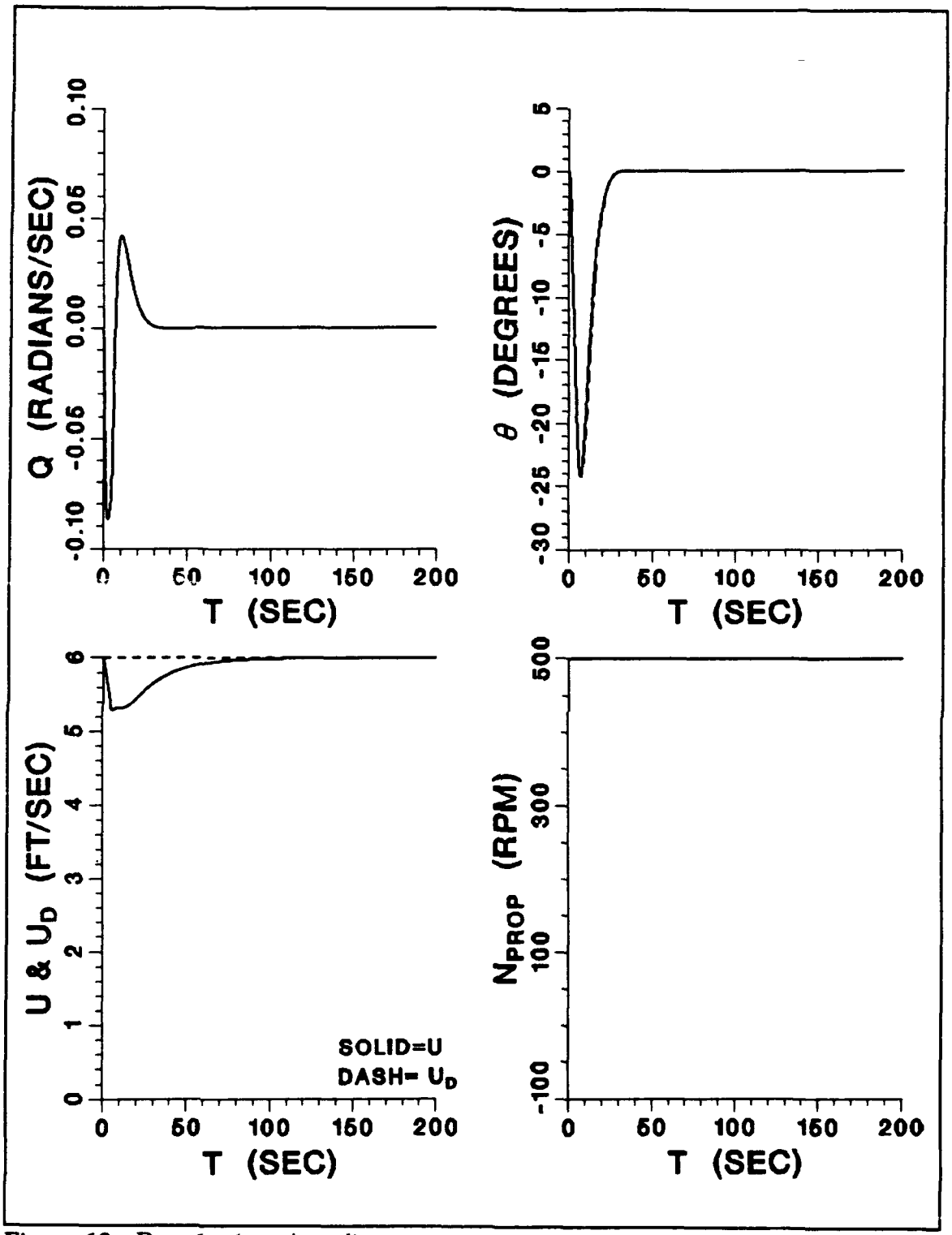


Figure 13. Run 6 - (continued)

hydrodynamic forces on the planes at higher speeds. The dominance of the stern planes can be seen on the graph of planes versus time in Figure 12. The stern planes actually shift over to a negative deflection as ordered depth is approached to reduce the pitch angle and eliminate overshoot.

The heave velocity w plays an insignificant role in reaching z_d. In fact, the AUV achieves a much smaller heave velocity in the cruise region as compared to the LQR controller in the hover region. Instead, the AUV uses pitch angle and surge velocity to reach ordered depth. In Figure 13 at u = 6.00 ft/s, we can see the AUV reaches a diving pitch angle of about -24 degrees.

E. CONTROL OF THE AUV IN THE TRANSITION REGION

1. State Space Equations

As discussed earlier, both planes and thrusters will be used in the transition region due to their comparable effects in this speed range. With all four inputs utilized, the control distribution matrix $\boldsymbol{B}$ at u = 1.250 ft/s is:

$$\boldsymbol{B} = \begin{bmatrix} -0.007959 & -0.017288 & 0.000166 & 0.000132 \\ 0.001672 & -0.008841 & -0.000047 & 0.000042 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{3.28}$$

Now, there are the same number of inputs as states, or in other words, n = m! This creates a severe problem. Utkin's method for designing a MIMO sliding mode controller requires the $\boldsymbol{B}_1$ matrix be nonsingular and of dimension m×m; hence, $\boldsymbol{B}_1$

must be a 4×4 matrix which makes it equivalent to $\boldsymbol{B}$! In this case, however, $\boldsymbol{B}_1$ is obviously singular and noninvertible:

$$\boldsymbol{B}_1 = \begin{bmatrix} -0.007959 & -0.017288 & 0.000166 & 0.000132 \\ 0.001672 & -0.008841 & -0.000047 & 0.000042 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{3.29}$$

The cause of the problem can be understood by examining the four equations that comprise the state space representation. There are only two equations, heave (3.1) and pitch (3.2), that contain the control inputs. The other two equations, (3.4) and (3.5), are kinematic relations which do not assist in the determination of the inputs. Thus, there are four unknowns - I_{bv}, I_{sv}, δ_{bp}, and δ_{sp} - and only two equations. The control inputs cannot be uniquely determined and an optimization scheme is needed. This was actually done in the hover and cruise regions by reducing the number of inputs from four to two based on their effectiveness, i.e., thrusters when hovering and planes when cruising.

In the transition region the effectiveness of the control inputs are essentially equivalent, and the number of inputs can be reduced by assuming a relationship between them. Here it was assumed:

$$\begin{aligned} I_{sv} &= I_{bv} \\ \delta_{sp} &= -\delta_{bp} \end{aligned} \tag{3.30}$$

and the linear state space representation at u = 1.25 ft/s becomes:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -0.0757 & -0.1684 & 0.0251 & 0.0 \\ 0.0188 & -0.2091 & -0.0668 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & -1.2500 & 0.0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} 0.00933 & 0.00030 \\ 0.01051 & -0.00001 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \delta_{bp} \\ I_{bv} \end{bmatrix} \quad (3.31)$$

The LQR technique can now be applied to design a MIMO sliding mode controller.

2. Controller Design by LQR Technique

After some experimentation, the LQR minimization matrix Q used in the cruise region was again used in the transition region, with $Q_{11} = 100$, $Q_{22} = 100$, $Q_{33} = 100$, and $Q_{44} = 1$. The controller equations become:

$$\begin{aligned} \sigma_1 &= 1.0\,w - 0.0735\,\theta + 0.0678\,(z - z_d) \\ \sigma_2 &= 1.0\,q + 1.0855\,\theta - 0.0735\,(z - z_d) \end{aligned} \quad (3.32)$$

$$\begin{aligned} \delta_{bp} &= 5.1\,w - 81.8\,q - 2.3\,\theta \\ &\quad - 1.6\,\eta^2\,satsgn(\sigma_1) - 93.7\,\eta^2\,satsgn(\sigma_2) \\ \delta_{sp} &= -\delta_{bp} \\ I_{bv} &= -134.7\,w + 3374.7\,q + 271.2\,\theta \\ &\quad - 3309.5\,\eta^2\,satsgn(\sigma_1) + 2937.0\,\eta^2\,satsgn(\sigma_2) \\ I_{sv} &= I_{bv} \end{aligned} \quad (3.33)$$

and the two nonzero poles are -0.1614 and -0.9919.

3. Computer Simulation with LQR Controller

Figures 14 to 17 show simulations of the AUV as generated by SDVDIVPPTTPSF_LQR.FOR and DPLOT2.FOR at two different speeds in the

transition region. Figures 14 and 15 show Run 7 at u = 0.90 ft/s, and Figures 16 and 17 show Run 8 at u = 1.60 ft/s. Both runs were executed with ϕ = 2.0 and η^2 = 0.2. The figures establish the controller works, if not spectacularly. This marginal performance is caused by the changing hydrodynamic forces on the planes in the transition region. At speeds above u = 1.25 ft/s, the planes begin to dominate. At lower speeds, the thrusters begin to dominate. The thrusters work together as they did in the hover region, while the planes deflect oppositely to create a pitch angle as they did in the cruise region.

a. Modified LQR Controller

The above scheme works well at the bow; the bow planes and bow thruster always work together. On the other hand, the stern is a problem because the stern planes and stern thruster always fight each other. This detrimental effect can be rectified by turning off the stern thruster at high speeds (above 1.25 ft/s) and by centering the stern planes at low speeds (below 1.25 ft/s).

Computer program SDVDIVPPTTPSF_LQR.FOR was altered to enact this modified controller and Figures 18 to 21 represent the results of Runs 9 and 10 using this controller. Note Figures 18 to 21 are labeled, "MODIFIED". Since all the set points are identical, Runs 7 and 8 can be easily compared to Runs 9 and 10. The modified controller is clearly superior, especially at the slower speed.

By modifying the controllers, the transition region has in effect been split into two regions; in essence there are a total of four control laws. If time permitted another iteration, a more elegant solution would be use the new

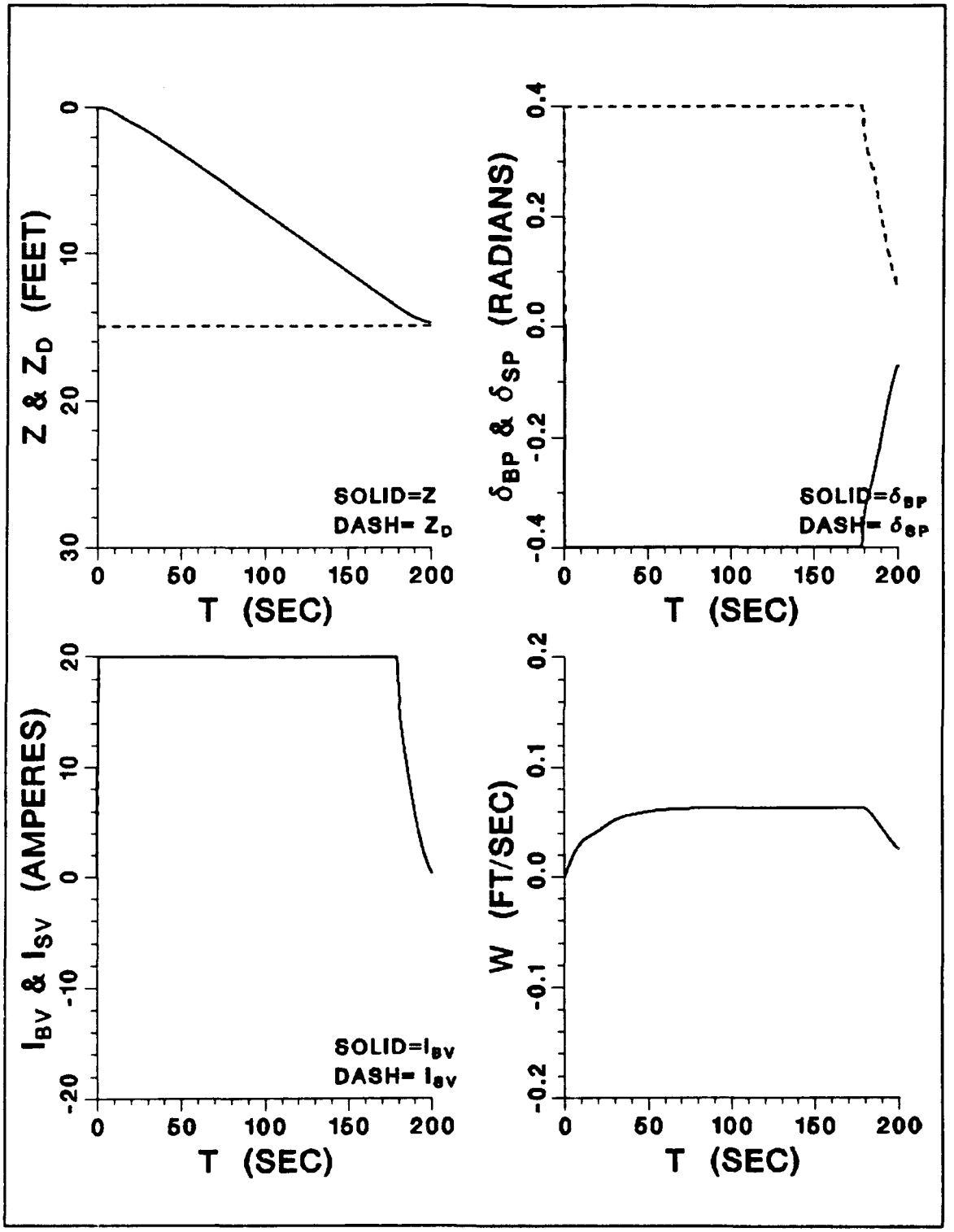


Figure 14. Run 7 - AUV Response in the Dive Plane at Transition Speed, LQR (u = 0.90 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

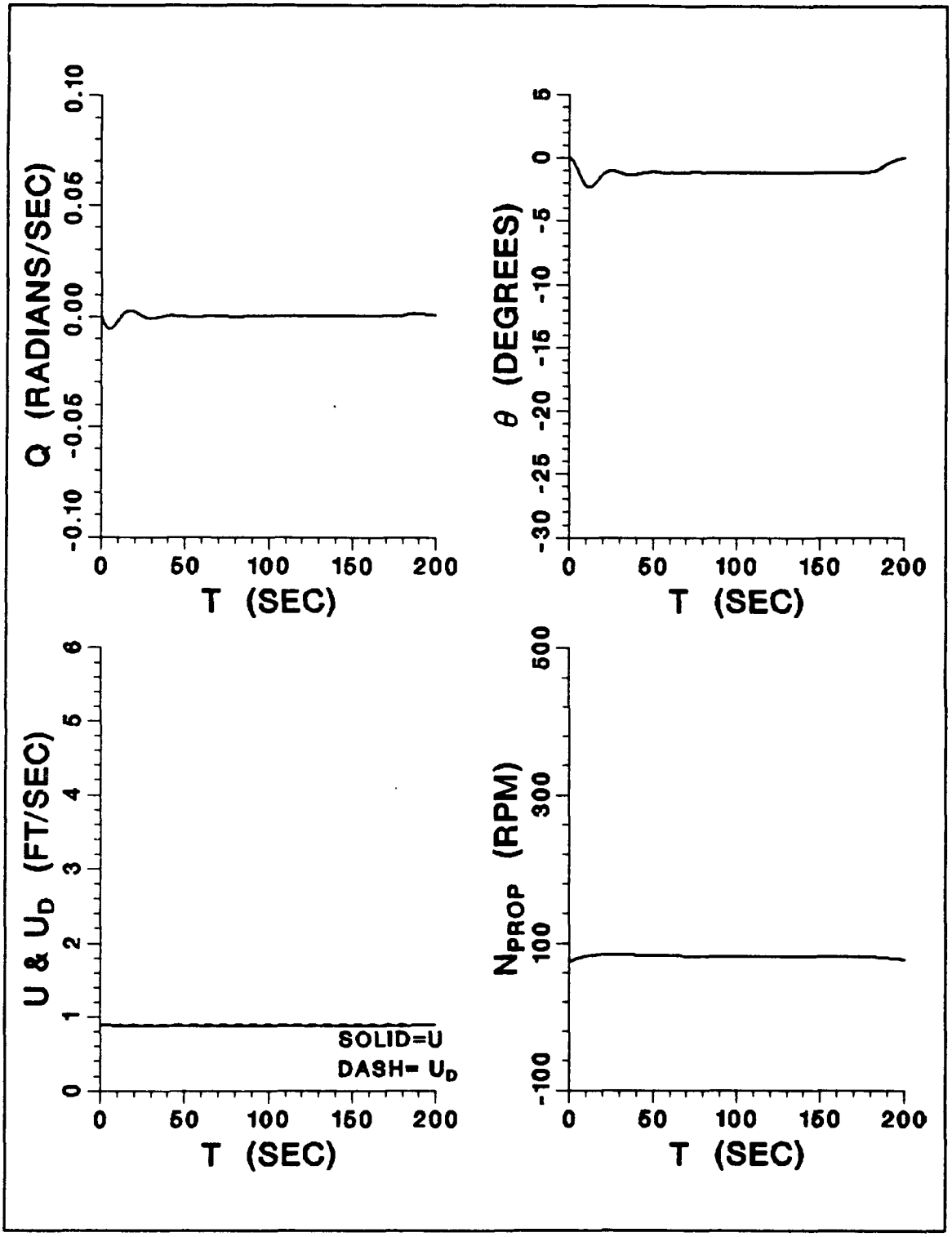


Figure 15. Run 7 - (continued)

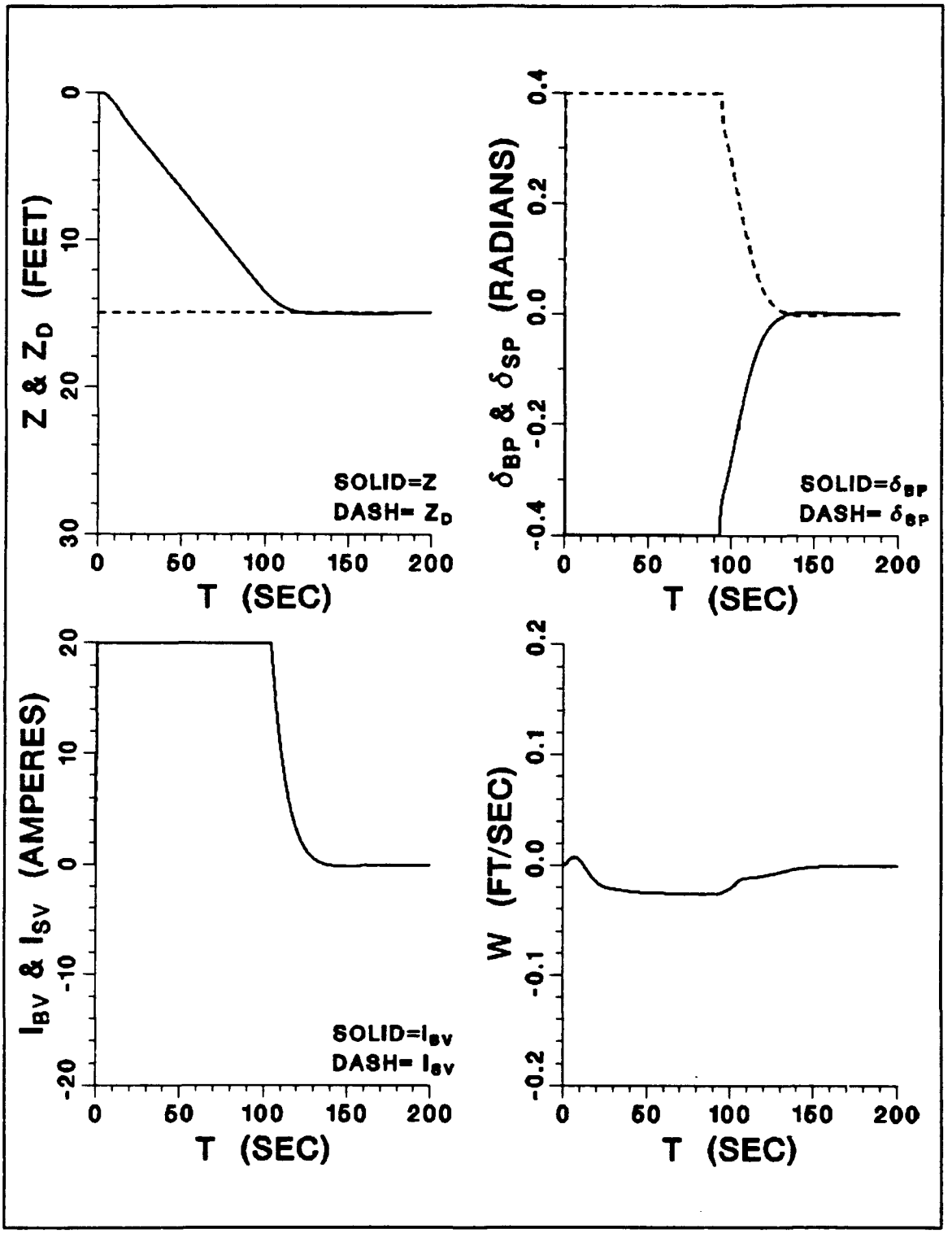


Figure 16. Run 8 - AUV Response in the Dive Plane at Transition Speed, LQR (u = 1.60 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

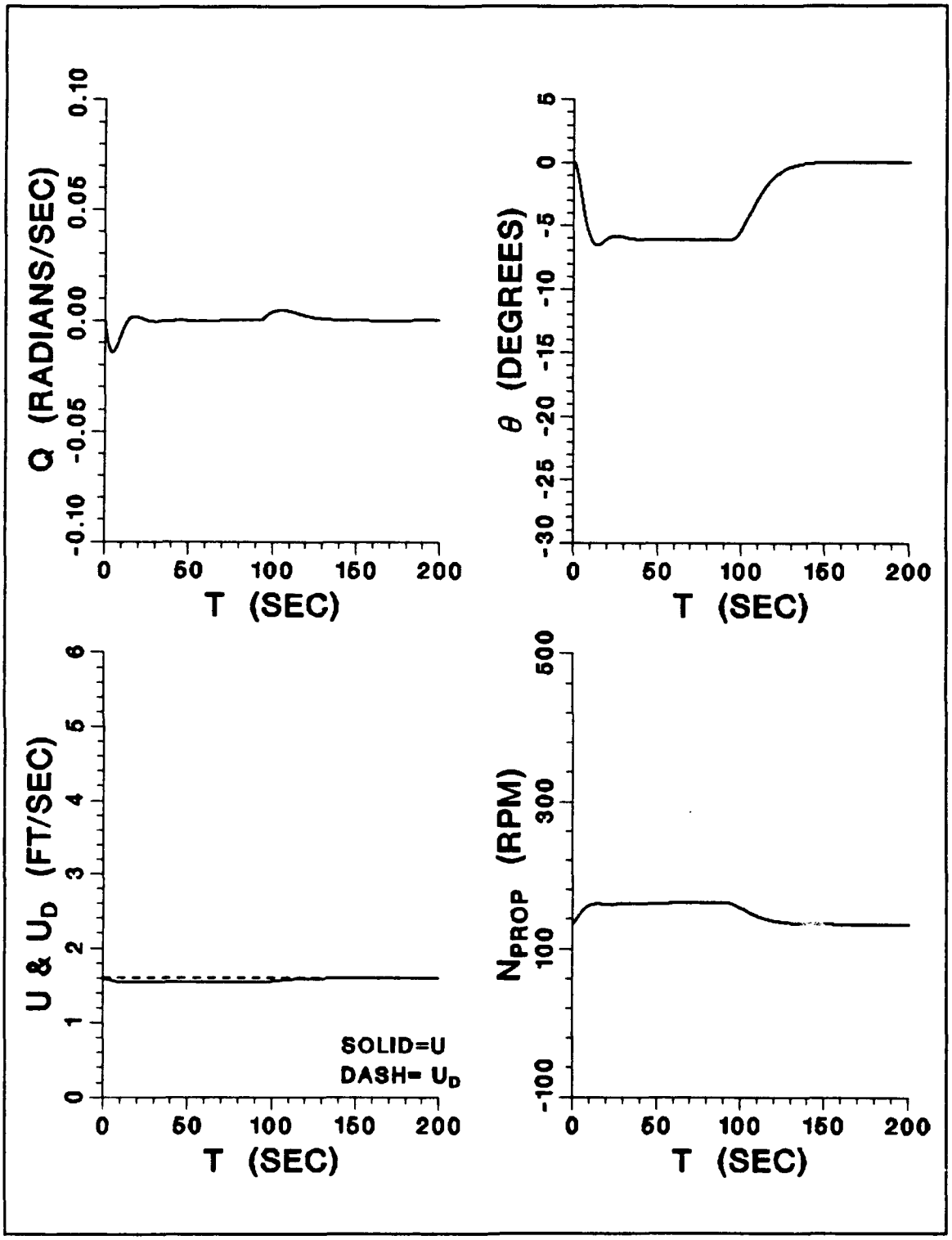


Figure 17. Run 8 - (continued)

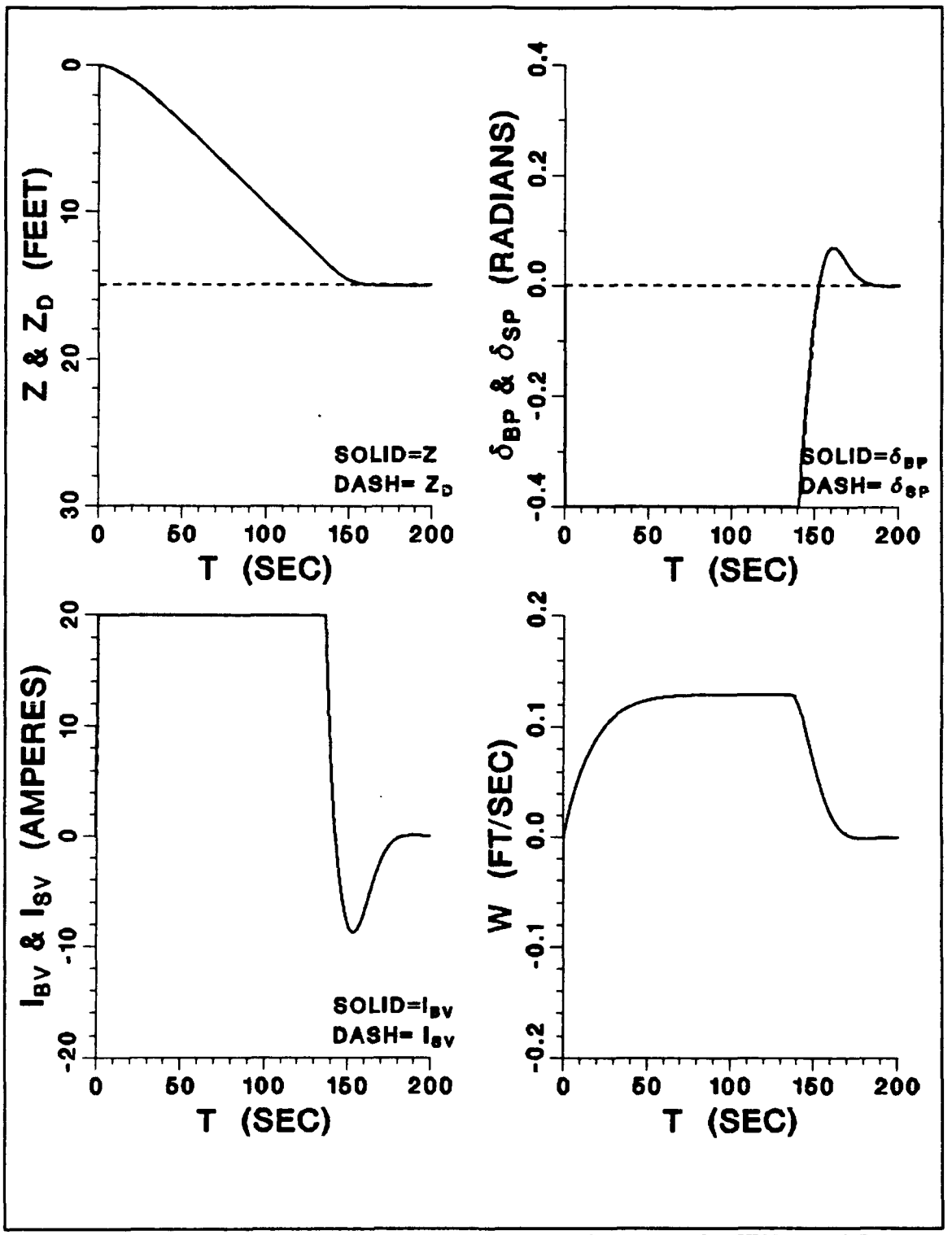


Figure 18. Run 9 - Diving Response at Transition Speed, MODIFIED LQR (u = 0.90 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

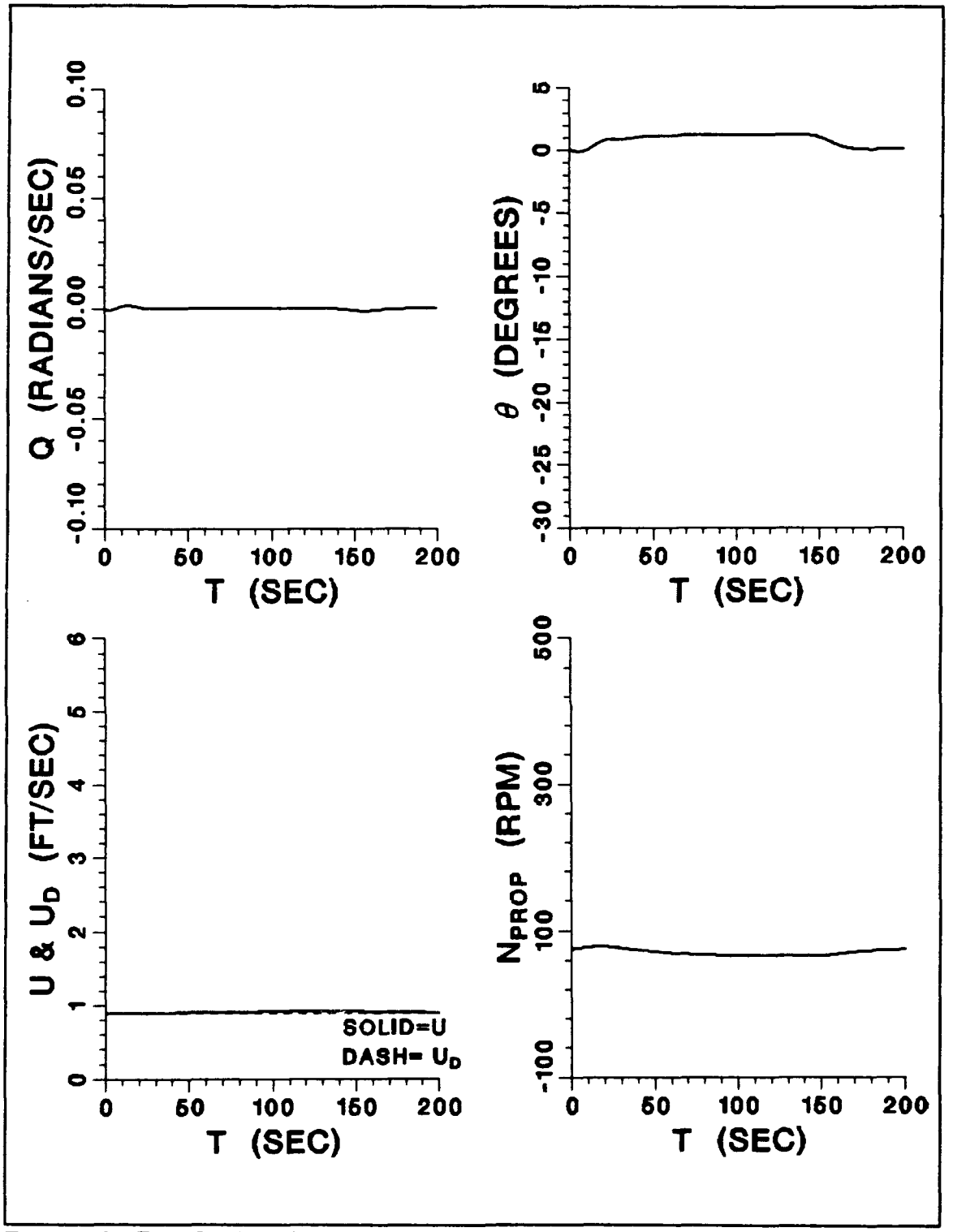


Figure 19. Run 9 - (continued)

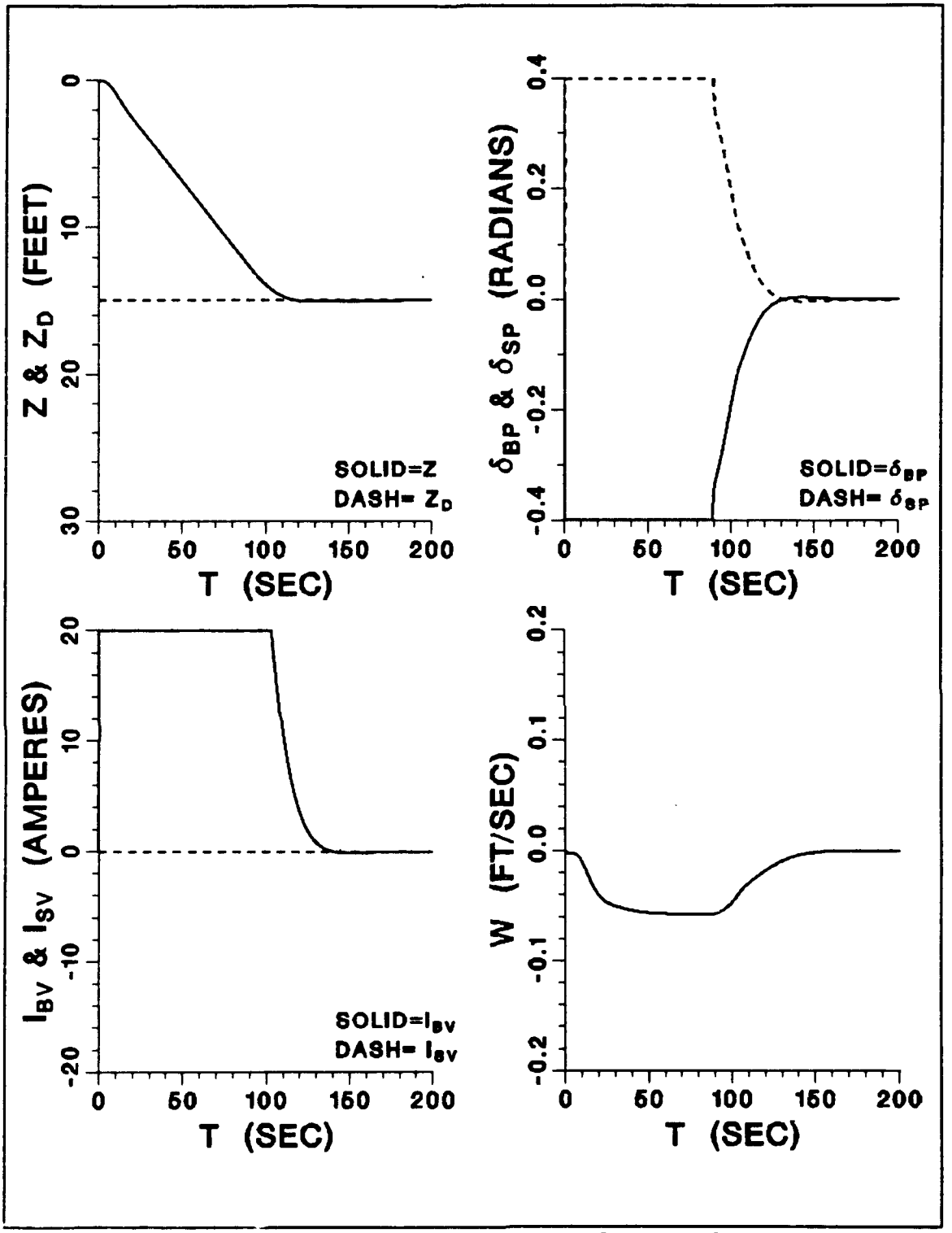


Figure 20. Run 10 - Diving Response at Transition Speed, MODIFIED LQR (u = 1.60 ft/s, ϕ = 2.0, η^2 = 0.2, Perfect State Feedback)

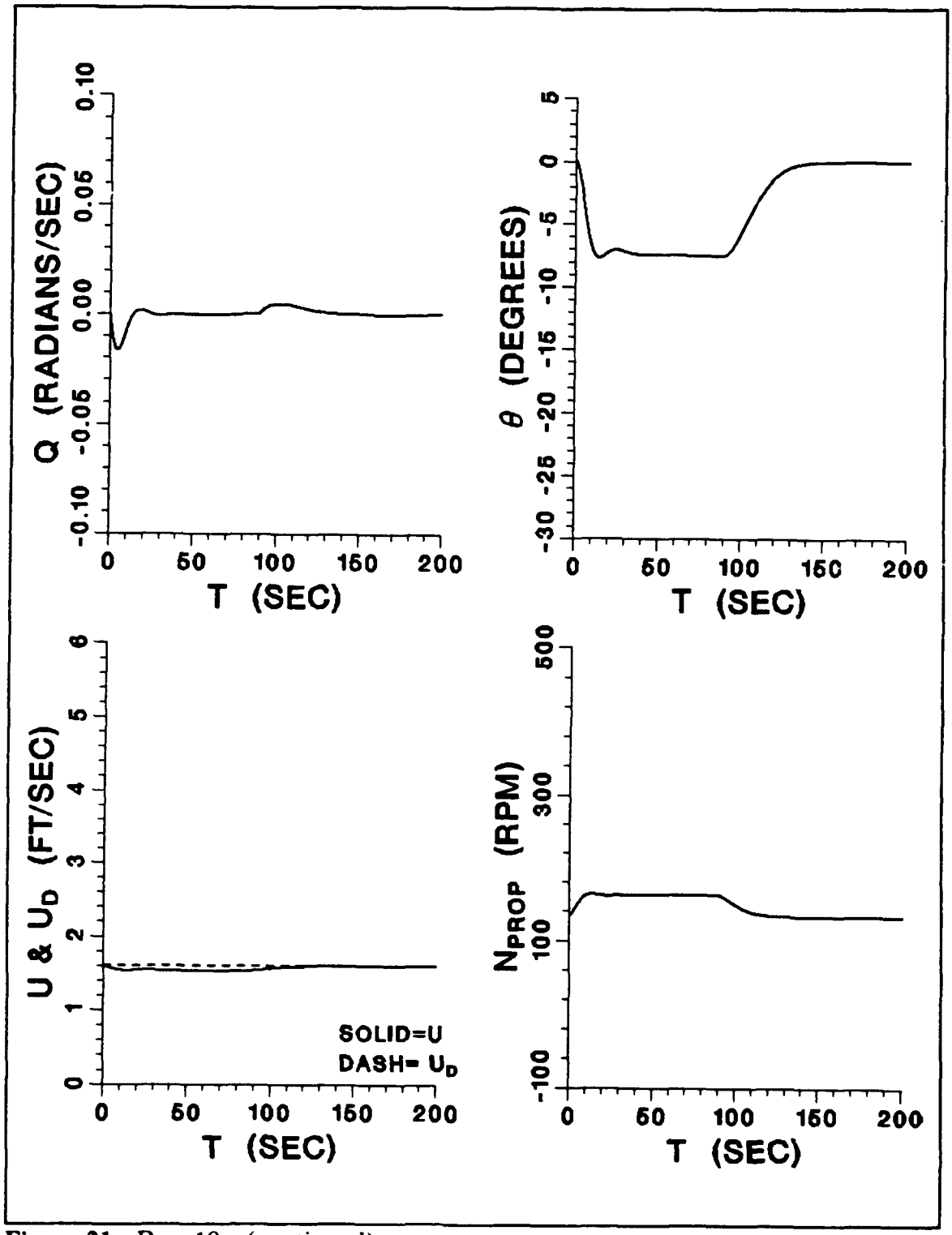


Figure 21. Run 10 - (continued)

assumptions to redesign two separate controllers in the transition region, one for speeds above 1.25 ft/s and one for speeds less than 1.25 ft/s. Theoretically the effectiveness of the controllers will be enhanced if surge velocity is divided into increasingly smaller regions, each with unique sliding surfaces and control laws. However, experience dictates that this scheme cannot be taken too far. The computer code cannot be made so large that it is unable to control the AUV in real time without unnecessary computational delays.

The control of the AUV in the transition region is difficult and the only consolation is the real vehicle will be operated in this region only for short periods of time as it speeds up to cruise or slows to hover.

F. OBSERVERS

1. Design

In addition to the chattering problem discussed in Chapter II, the requirement of full state feedback is a major disadvantage of the sliding mode control technique because in most cases one or more of the states cannot be directly measured. Up to this point, it has been assumed the controllers have perfect state feedback. In fact, the real AUV has no sensor for measuring the heave velocity w. The predicament is easily overcome by designing a reduced order (Luenberger) observer to estimate the heave velocity w. The method of designing a Luenberger observer is well known and will not be discussed in detail here; however, a couple of interesting points need to be discussed.

The observers are designed starting with the governing equations expressed in the linear state space representation of (2.1), restated here:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3.34}$$

The $\boldsymbol{A}$ matrix is the same for all three speed regions if it is expressed as a function of surge velocity u:

$$\boldsymbol{A} = \begin{bmatrix} a_{11}u & a_{12}u & a_{13} & 0.0 \\ a_{21}u & a_{22}u & a_{23} & 0.0 \\ 0.0 & a_{32} & 0.0 & 0.0 \\ a_{41} & 0.0 & a_{43}u & 0.0 \end{bmatrix} \tag{3.35}$$

Thus, this $\boldsymbol{A}$ matrix can be used for all three regions; however, this cannot be done for the $\boldsymbol{B}$ matrix because the inputs $\boldsymbol{u}$ are different in each region. If $\boldsymbol{B}$ is expressed as a function of surge velocity u for each region, they can be written as:

- Hover region:

$$\boldsymbol{B} = \begin{bmatrix} b_{11}(0.25 - 0.008\bar{3}u) & b_{12}(0.25 - 0.008\bar{3}u) \\ b_{21}(0.25 - 0.008\bar{3}u) & b_{22}(0.25 - 0.008\bar{3}u) \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} I_{bv} \\ I_{sv} \end{bmatrix} \tag{3.36}$$

- Transition region:

$$\boldsymbol{B} = \begin{bmatrix} b_{11}u^2 & b_{12}(0.25 - 0.008\bar{3}u) \\ b_{21}u^2 & b_{22}(0.25 - 0.008\bar{3}u) \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \delta_{bp} \\ I_{bv} \end{bmatrix} \tag{3.37}$$

- Cruise region:

$$\boldsymbol{B} = \begin{bmatrix} b_{11}u^2 & b_{12}u^2 \\ b_{21}u^2 & b_{22}u^2 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \delta_{bp} \\ \delta_{sp} \end{bmatrix} \tag{3.38}$$

Expressing $\boldsymbol{B}$ in general terms:

$$\boldsymbol{B} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \tag{3.39}$$

Then the observer equations can also be expressed in general for all three observers:

$$\hat{w} = L_1 q + L_2 \theta + L_3(z - z_d) + \hat{Z} \tag{3.40}$$

$$\mathring{\hat{Z}} = F\hat{Z} + G_1 q + G_2 \theta + G_3(z - z_d) + H_1 u_1 + H_2 u_2 \tag{3.41}$$

$$F = a_{11}u - L_1 a_{21}u - L_3 a_{41} \tag{3.42}$$

$$G_1 = a_{12}u - L_1 a_{22}u - L_2 a_{32} + FL_1,\ G_2 = a_{13} - L_1 a_{23} - L_3 a_{43}u + FL_2,\ G_3 = FL_3 \tag{3.43}$$

$$H_1 = B_{11} - L_1 B_{21} \ , \ H_2 = B_{12} - L_1 B_{22} \tag{3.44}$$

$$L_1 = \frac{a_{11}u - S_1}{a_{21}u + a_{41}} \ , \ L_2 = 0.0 \ , \ L_3 = L_1 \tag{3.45}$$

$$u_1 \equiv input\ 1 \ , \ u_2 \equiv input\ 2 \tag{3.46}$$

$$S_1 \equiv observer\ pole \tag{3.47}$$

2. Simulation with Observers

The LQR simulation program was rewritten with observers for the heave velocity *w*. The new program is called SDVDIVPPTTOBS_LQR.FOR and can be found in Appendix C. The three observers are nested in subroutines at the end of the program. DISSPLA plotting program DPLOT4.FOR (see Appendix A) was used to generate graphs of simulation runs.

To allow the observers to settle out after the start of the simulation, a 10 second delay was programmed into all the observers during which the estimated value of *w* was set to zero. The pole of each observer was set to a value twice as fast as the slowest sliding surface pole in the respective speed region.

Figures 22 to 27 show simulation Runs 11 to 13 that were generated by the programs. Each run was conducted in a different controller speed region. The figures reveal that the observers work well. Vehicle control is not compromised by using an estimated value for the heave velocity.

It is imperative that an observer for the heave velocity is utilized at slower speeds because *w* becomes the predominant value used by the controller. In fact, it

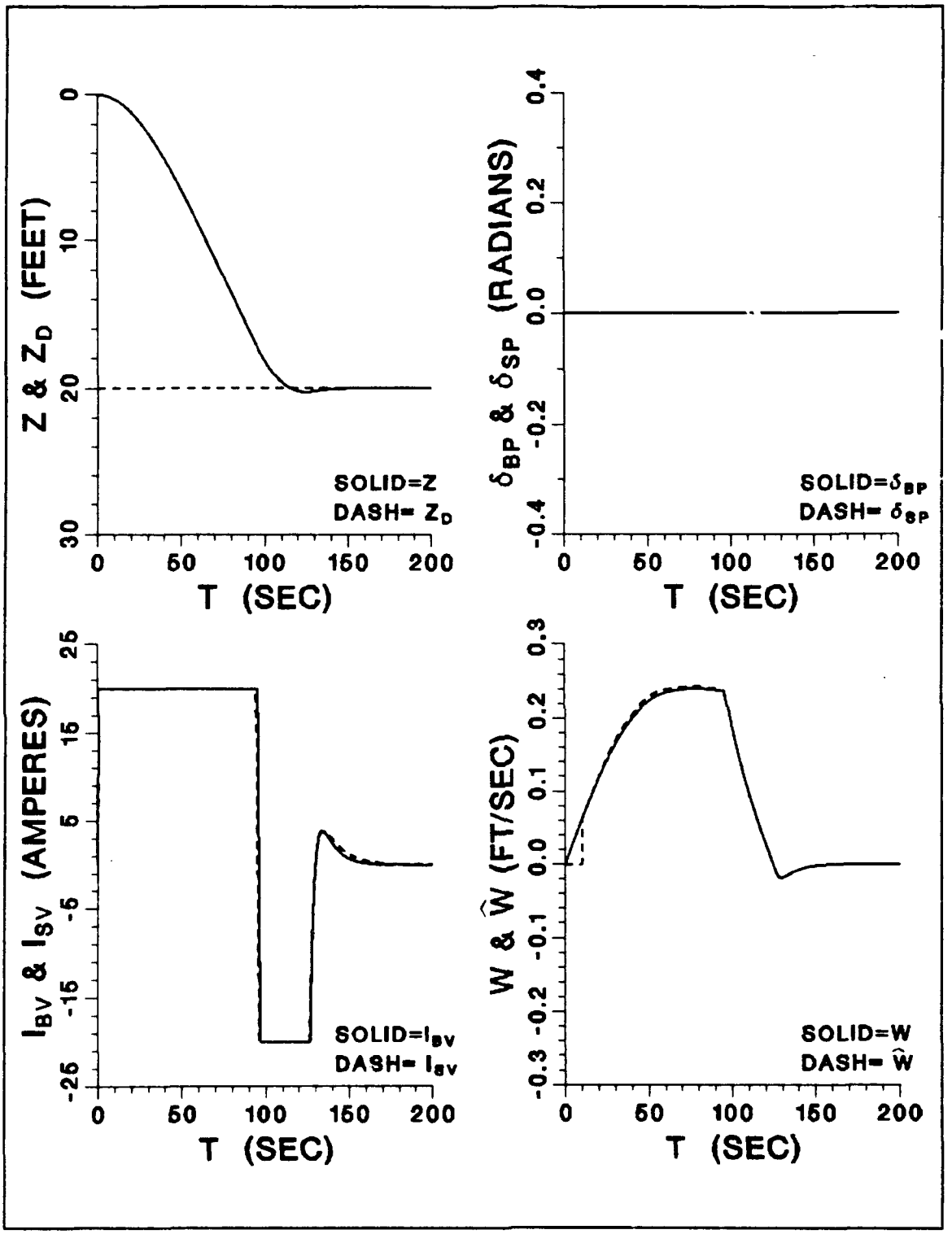


Figure 22. Run 11 - AUV Response in the Dive Plane at Hover Speed, LQR (u = 0.00 ft/s, ϕ = 2.0, η^2 = 0.2, w Observed)

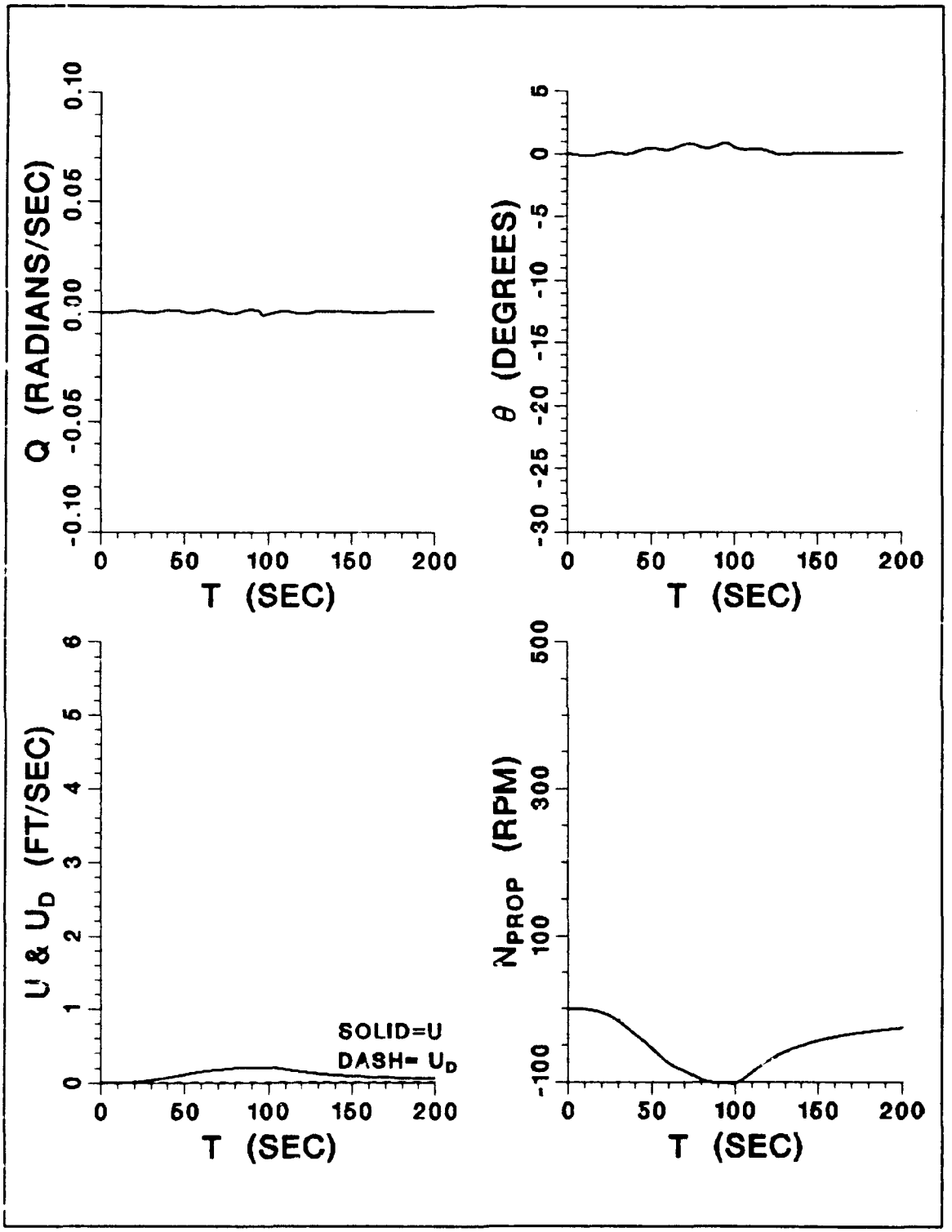


Figure 23. Run 11 - (continued)

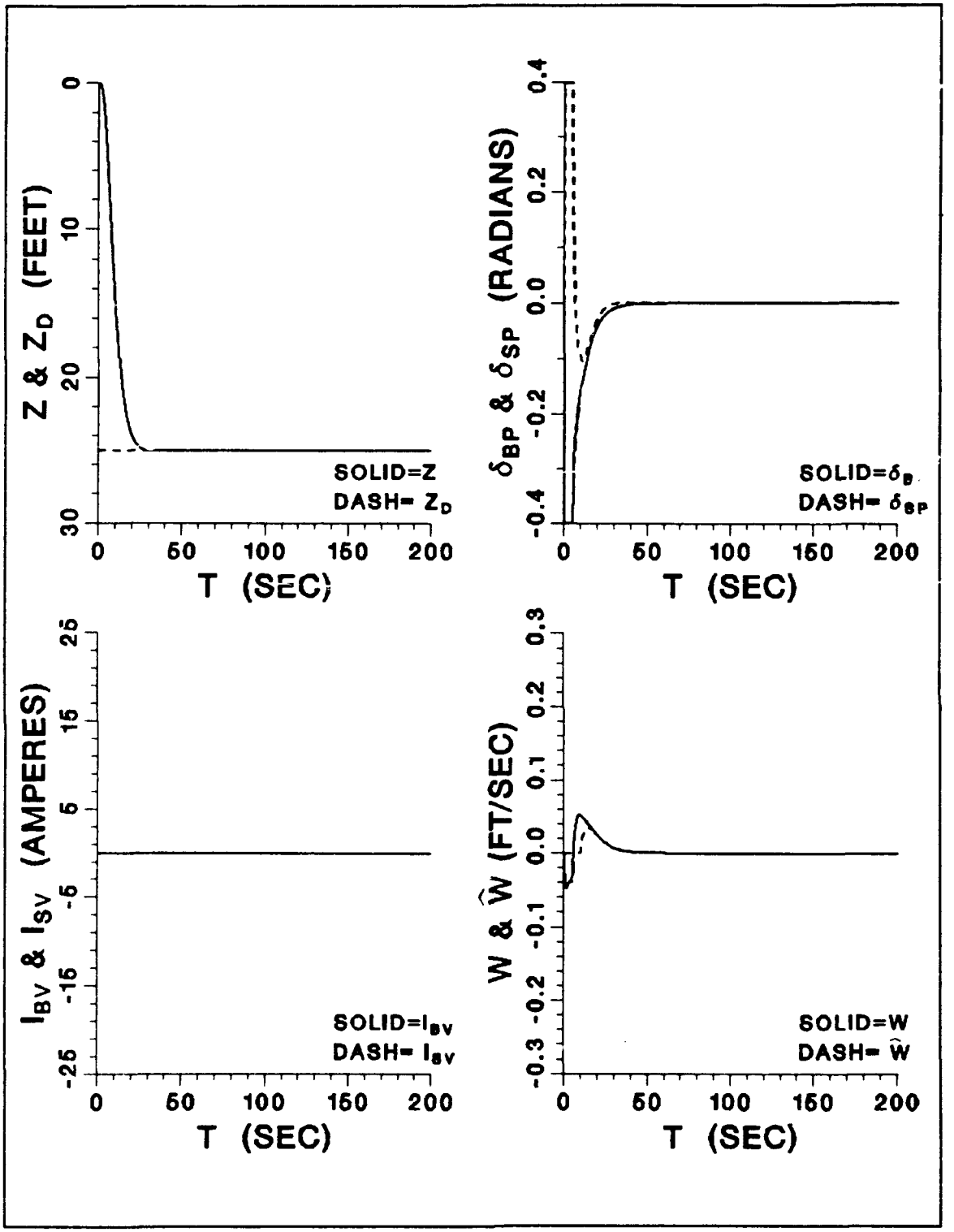


Figure 24. Run 12 - AUV Response in the Dive Plane at Cruise Speed, LQR (u = 6.00 ft/s, ϕ = 1.0, η^2 = 0.35, w Observed)

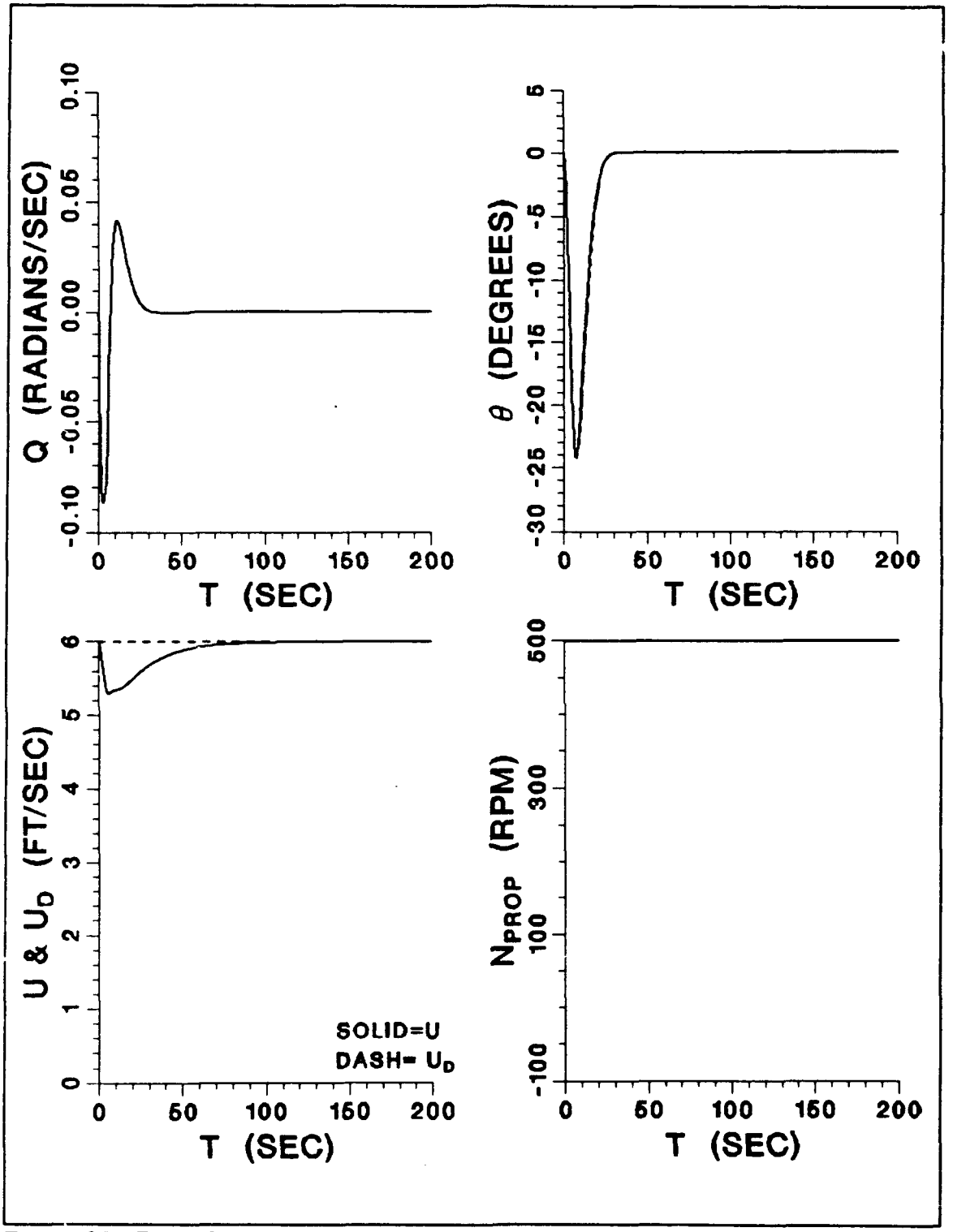


Figure 25. Run 12 - (continued)

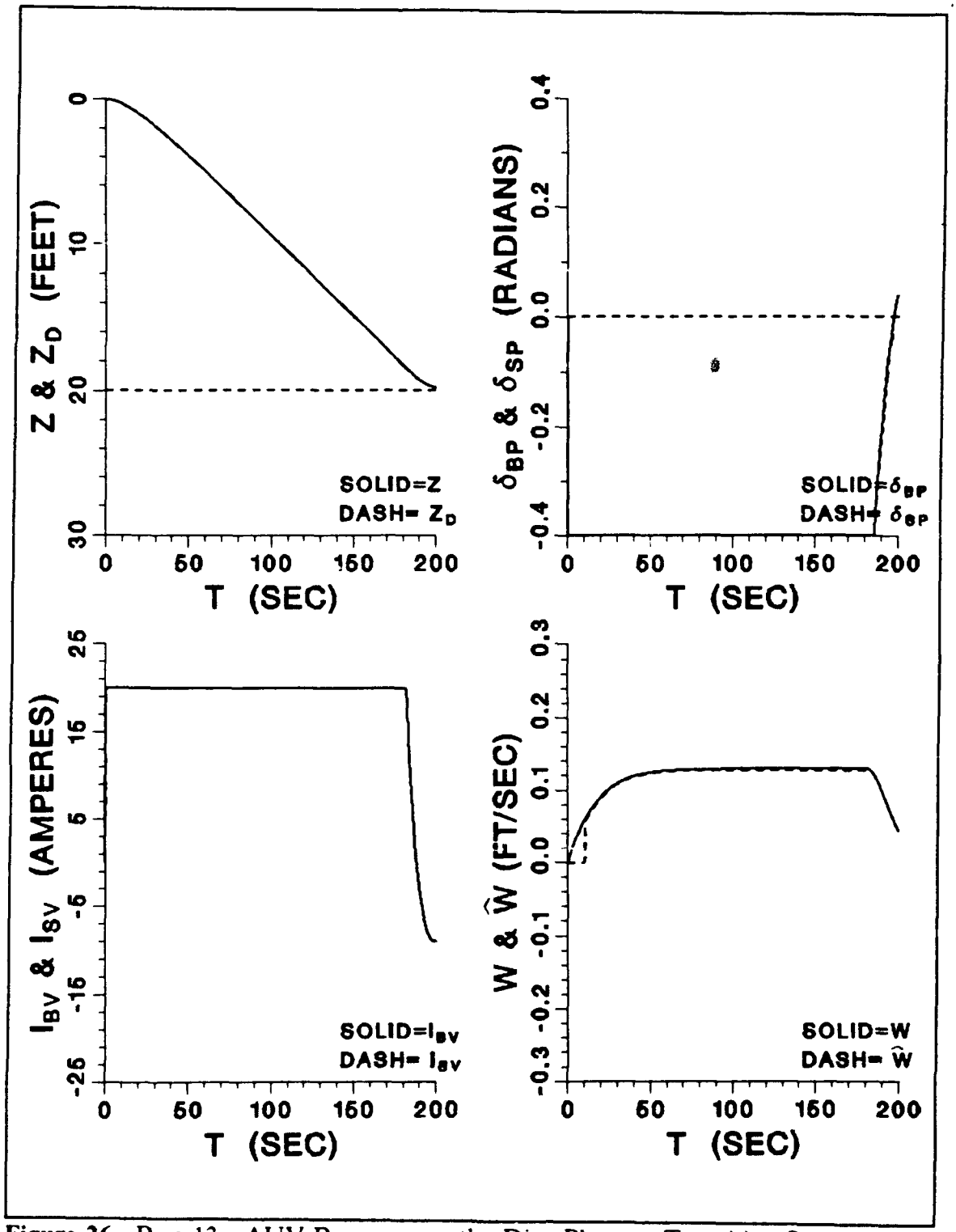


Figure 26. Run 13 - AUV Response in the Dive Plane at Transition Speed, LQR (u = 0.90 ft/s, ϕ = 2.0, η^2 = 0.2, w Observed)

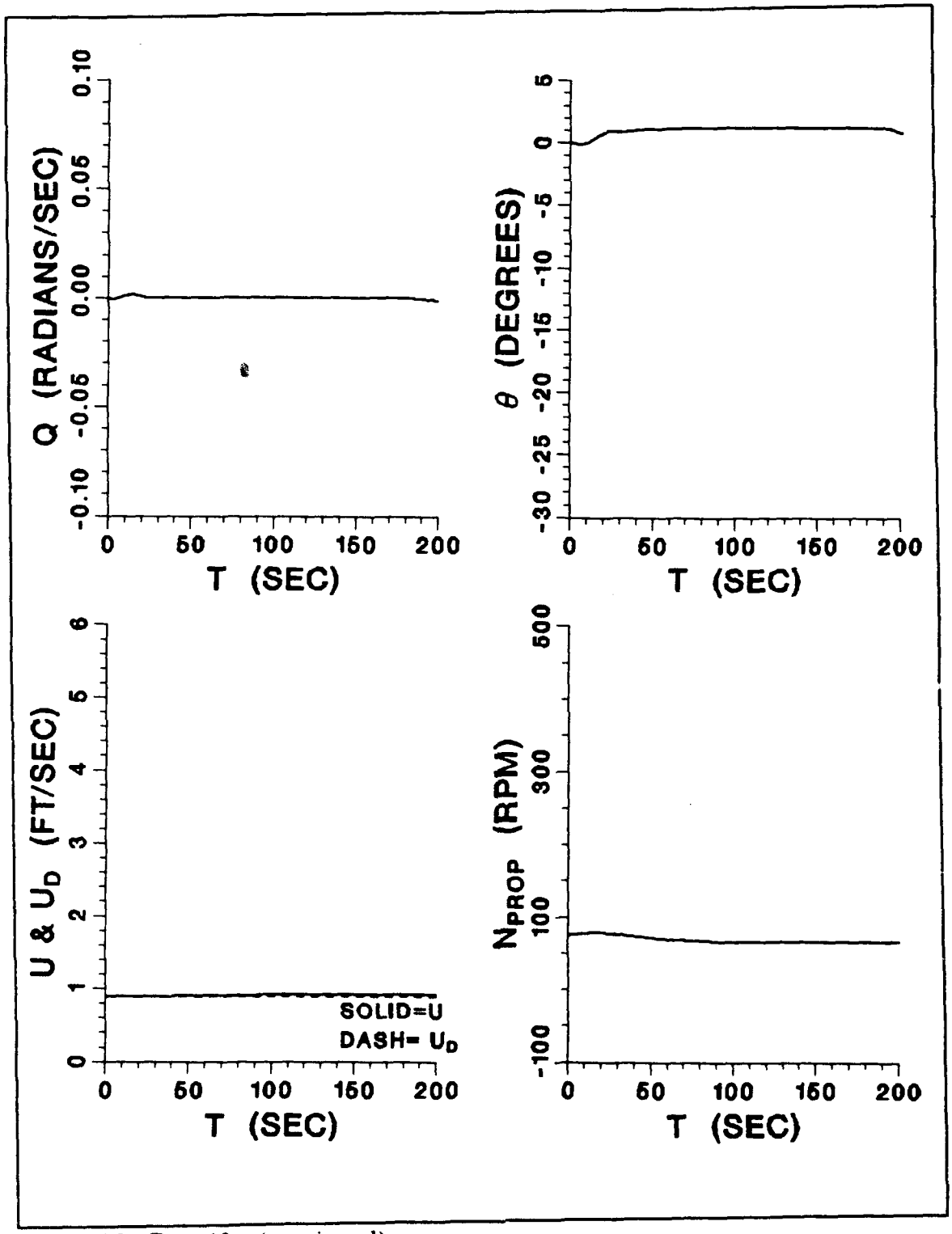


Figure 27. Run 13 - (continued)

was found that an observer was not needed in the cruise region because w is inconsequential at high speeds.

G. TRANSITION IN/OUT OF HOVER

One of the most difficult maneuvers for an AUV is the transition into or out of hover. This is because control of the vehicle is passing between controllers and observers. Program SDVDIVPPTTOBS_LQR.FOR was modified for differing initial conditions to explore this maneuver. Figures 28 and 29 show Run 14 in which the AUV transitions into hover while conducting a depth change. This is a radical maneuver. Figures 30 and 31 show Run 15 in which the AUV transitions out of hover while conducting a depth change.

The figures exhibit the resilience of the controllers and observers. The radical maneuvers create disturbances on the vehicle, but they are overcome by the control package and the vehicle settles out. In Run 14, there is some oscillation of the vehicle around 0 degree of pitch as it is thrusted to ordered depth; however, the oscillation is only about $\pm$ 4 degrees and is acceptable.

H. CONCLUDING REMARKS

A controller package has been designed for the AUV that effectively controls the vehicle throughout its entire operational speed range. The controller package has been optimized by dividing up the surge velocity into three different regions and designing a separate controller and observer for each region. This strategy is necessary because the hydrodynamic forces on the planes change with surge velocity.

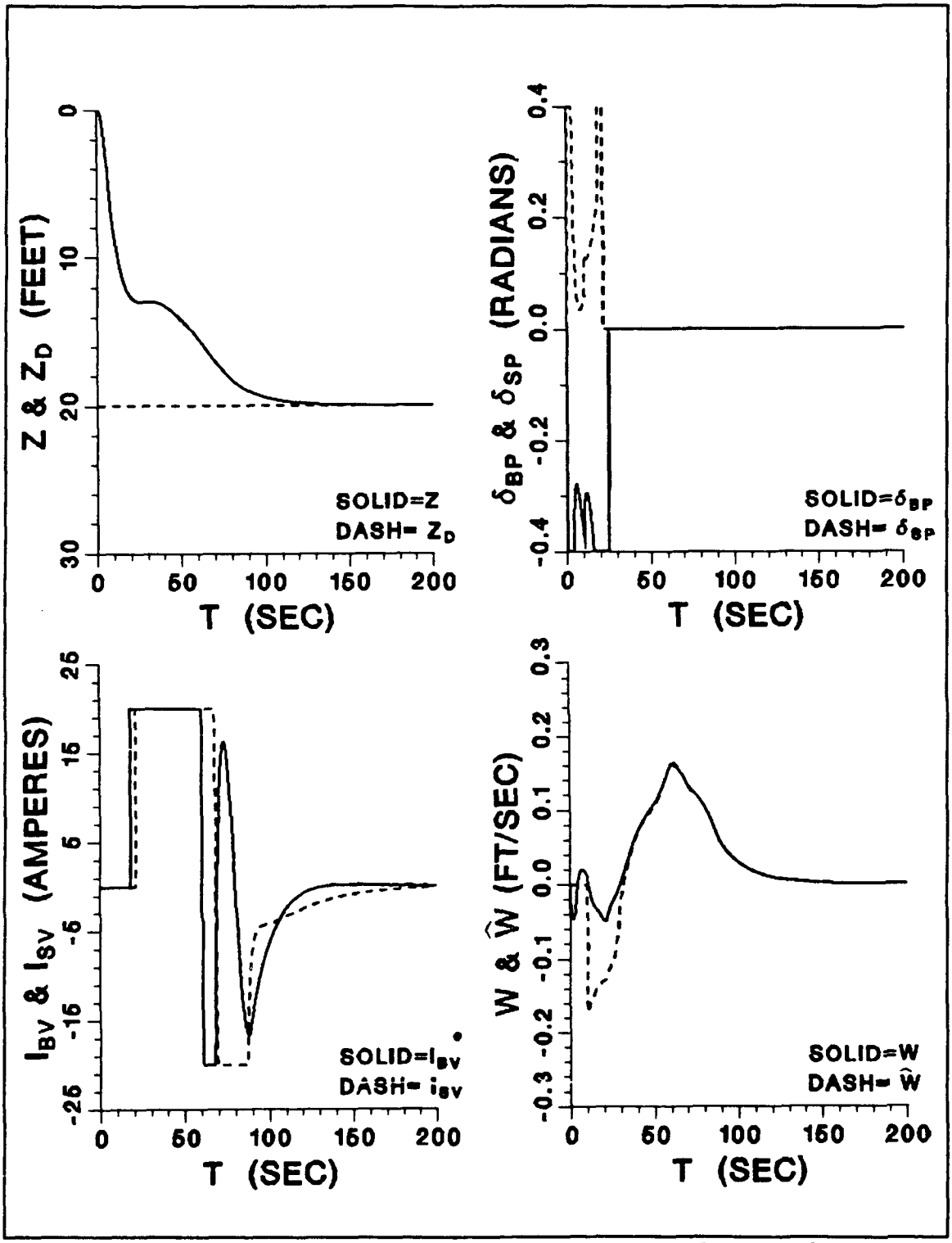


Figure 28. Run 14 - AUV Response While Transitioning to Hover, LQR (ϕ = 2.0, η^2 = 0.2, w Observed)

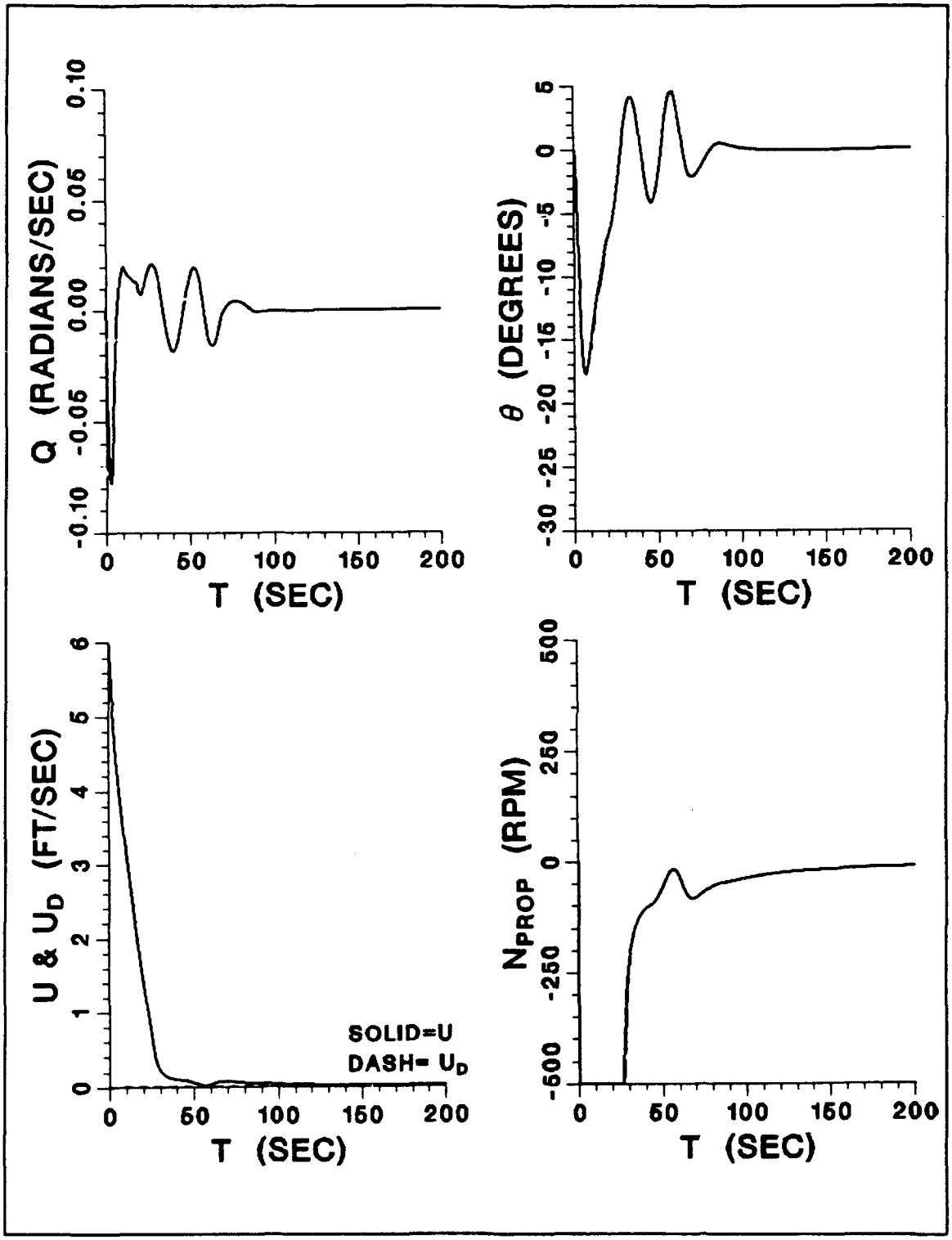


Figure 29. Run 14 - (continued)

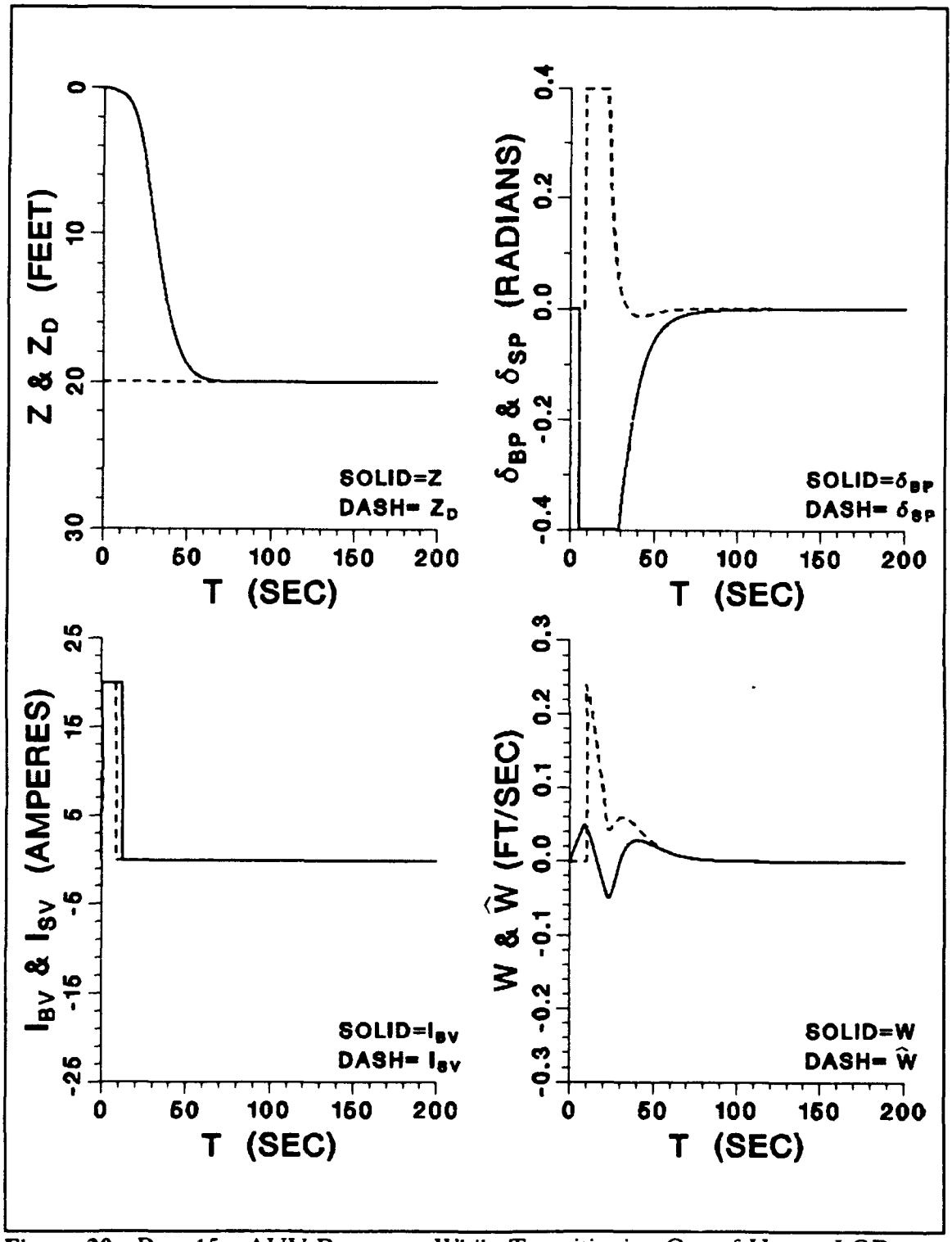


Figure 30. Run 15 - AUV Response While Transitioning Out of Hover, LQR (ϕ = 2.0, η^2= 0.2, w Observed)

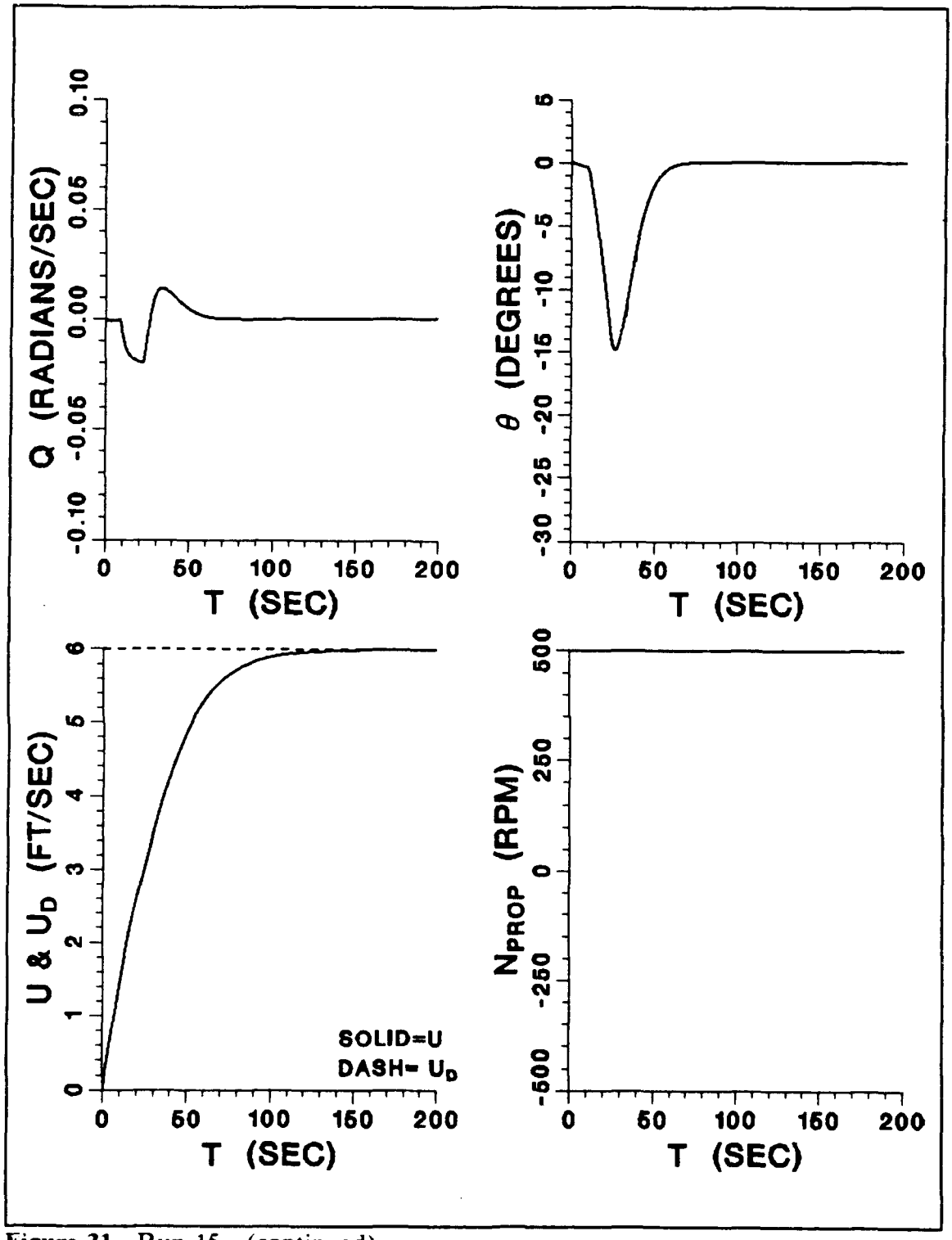


Figure 31. Run 15 - (continued)

Obviously to enact a MIMO sliding mode controller for the real NPS AUV, it will be necessary to verify all hydrodynamic coefficients. In addition, Kalman filters will have to be used on the real vehicle to smooth the measurements of the states. This due to noise.

IV. LINE-OF-SIGHT GUIDANCE OF THE AUV USING MIMO SLIDING MODE CONTROL

A. INTRODUCTION

Control of the AUV in the horizontal plane will now be analyzed. Steering control of the AUV is essentially analogous to depth control, with minor excursions. The first difference is the predominance of disturbances in the horizontal plane, especially ones of a steady-state nature. Transient disturbances are easily handled by the inherent robustness of the sliding mode control design.

In the dive plane, environmental steady-state disturbances are uncommon; ocean currents are mainly horizontal. Of course, transient disturbances due to wave action occur near the surface. However, the effect of ocean currents on the AUV requires steady-state disturbances to be accounted for in steering controller design.

Another difference of steering control is the requirement to adhere to a navigation scheme. Two common methods of navigation are line-of-sight (LOS) and cross-track-error (CTE) guidance. This chapter deals with LOS guidance, while the latter is discussed in Chapter V. LOS guidance of the NPS AUV was first investigated by Lienard [Ref. 3] for SISO systems.

For two reasons, this thesis will investigate steering control at higher speeds by the use of rudders only. First, time and space requirements for this thesis don't permit analyzing the horizontal plane in the same manner as the dive plane - at three

different speed ranges. Second, and more importantly, once MIMO steering with rudders has proven effective for guidance control, it would be unnecessarily repetitive to repeat the analysis with horizontal tunnel thrusters. Obviously, the three tier control hierarchy used successfully for diving can be extended to steering.

B. NONLINEAR GOVERNING EQUATIONS IN THE HORIZONTAL PLANE

1. Addition of Bow Rudder to SDV Model

As discussed in Chapter III, a bow rudder needs to be added to the SDV model of [Ref. 7] to approximate the geometry of the NPS AUV. The assumption was made that the bow rudder would have the same effect on the vehicle as the "real" stern rudder. Hence, the magnitude of the hydrodynamic coefficients would be equal for the two rudders; although, in some cases they would be of opposite sign:

$$
\begin{aligned}
Y'_{\delta_{br}} &= 2.730\times10^{-2} & Y'_{\delta_{sr}} &= 2.730\times10^{-2} \\
N'_{\delta_{br}} &= 1.290\times10^{-2} & N'_{\delta_{sr}} &= -1.290\times10^{-2} \\
X'_{r\delta_{br}} &= 8.180\times10^{-4} & X'_{r\delta_{sr}} &= -8.180\times10^{-4} \\
X'_{v\delta_{br}} &= 1.730\times10^{-3} & X'_{v\delta_{sr}} &= 1.730\times10^{-3} \\
X'_{\delta_{br}\delta_{br}} &= -1.010\times10^{-2} & X'_{\delta_{sr}\delta_{sr}} &= -1.010\times10^{-2}
\end{aligned}
\qquad (4.1)
$$

2. Assumptions

For control of the vehicle in the horizontal plane, the following assumptions have been made:

- Vehicle motion is confined to the horizontal plane:

$$(p, \dot{p}, q, \dot{q}, \theta, \phi, w, \delta_{br}, \delta_{sr}, I_{bv}, I_{sv} = 0)$$

- The AUV is neutrally buoyant.
- The CB and CG are on the vehicle's centerline.
- The AUV is restricted to speeds in which the rudders are effective.
- The rudders have a restricted range of motion identical to the planes.
- Surge velocity can be assumed constant in the yaw and pitch equations.
- Rudders are infinitely responsive, i.e., no lag in reaction time. (This is a valid assumption because steering gear dynamics are much faster than the dynamics of a small turning vehicle.)

3. Resulting Equations

After all modifications and assumptions, the nonlinear governing equations in the horizontal plane from [Ref. 7] become:

- Lateral (Sway) equation of motion

$$\begin{aligned} m(\dot{v} + ur + x_g\dot{r} - y_g r^2) = \; & \frac{\rho}{2}L^4 Y'_{\dot{r}}\dot{r} \\ & + \frac{\rho}{2}L^3(Y'_{\dot{v}}\dot{v} + Y'_r ur) \\ & + \frac{\rho}{2}L^2[Y'_v uv + u^2(Y'_{\delta_{br}}\delta_{br} + Y'_{\delta_{sr}}\delta_{sr})] \\ & - \frac{\rho}{2}\int_{stern}^{bow} \frac{C_{Dy}h(x)(v + xr)^3}{U_{cf}(x)}dx \end{aligned} \tag{4.2}$$

- Yaw equation of motion

$$
\begin{aligned}
I_z\dot{r} + mx_g(\dot{v} + ur) + my_g(vr - \dot{u}) = \\
&\frac{\rho}{2}L^5 N'_{\dot{r}}\dot{r} \\
&+ \frac{\rho}{2}L^4(N'_{\dot{v}}\dot{v} + N'_{r}ur) \\
&+ \frac{\rho}{2}L^3[N'_{v}uv + u^2(N'_{\delta_{br}}\delta_{br} + N'_{\delta_{sr}}\delta_{sr})] \\
&- \frac{\rho}{2}\int_{\text{stern}}^{\text{bow}} \frac{C_{Dy}h(x)(v + xr)^3}{U_{cf}(x)}x\,dx
\end{aligned}
\tag{4.3}
$$

- Surge equation of motion

$$
\begin{aligned}
m(\dot{u} - vr - x_g r^2 - y_g\dot{r}) = \\
&\frac{\rho}{2}L^4 X'_{rr}r^2 \\
&+ \frac{\rho}{2}L^3[X'_{\dot{u}}\dot{u} + X'_{vr}vr + ur(X'_{r\delta_{br}}\delta_{br} + X'_{r\delta_{sr}}\delta_{sr})] \\
&+ \frac{\rho}{2}L^2[X'_{vv}v^2 + uv(X'_{v\delta_{br}}\delta_{br} + X'_{v\delta_{sr}}\delta_{sr})] \\
&+ \frac{\rho}{2}L^2u^2(X'_{\delta_{br}\delta_{br}}\delta_{br}^2 + X'_{\delta_{sr}\delta_{sr}}\delta_{sr}^2) \\
&+ \frac{\rho}{2}L^2u^2X'_{\text{prop}}
\end{aligned}
\tag{4.4}
$$

- Kinematic definition

$$
\dot{\psi} = r \tag{4.5}
$$

- Cross flow velocity equation

$$U_{cf}(x) = |v + xr| \tag{4.6}$$

- Inertial position rates

$$\begin{aligned} \dot{x} &= u\cos\psi - v\sin\psi \\ \dot{y} &= u\sin\psi + v\cos\psi \end{aligned} \tag{4.7}$$

The propulsion equations (3.6) to (3.10) still apply in the horizontal plane.

C. LINEAR STATE SPACE REPRESENTATION

The yaw and pitch equations are linearized at $u = 3.0$ ft/s to obtain the equations of motion in the linear state space representation of (2.1) for control law design. As in the dive plane, a Fortran program was utilized to assist in the solution of the yaw and sway equations (see Appendix A). With the inclusion of the linear kinematic definition (4.5), the state space representation is complete:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -0.1265 & -1.0547 & 0.0 \\ -0.0084 & -0.2952 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} 0.1168 & 0.1036 \\ 0.0398 & -0.0382 \\ 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} \delta_{br} \\ \delta_{sr} \end{bmatrix} \tag{4.8}$$

As before, (4.8) is partitioned into the form of (2.21) and (2.22) for the application of Utkin's MIMO sliding mode control design technique:

$$A_{11} = \begin{bmatrix} -0.1265 & -1.0547 \\ -0.0084 & -0.2952 \end{bmatrix} \qquad A_{12} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$$

$$A_{21} = [0.0 \quad 1.0] \qquad A_{22} = [0.0]$$

$$B_1 = \begin{bmatrix} 0.1168 & 0.1036 \\ 0.0398 & -0.0382 \end{bmatrix} \tag{4.9}$$

$$y_1 = [v \quad r]^T \qquad y_2 = [\psi] \qquad u = [\delta_{br} \quad \delta_{sr}]^T$$

D. CONTROLLER DESIGN BY LQR TECHNIQUE

Matrix-x was again utilized for calculations by the LQR design technique. After experimenting with the diagonal of the LQR minimization matrix Q, it was found that satisfactory results were obtained with $Q_{11} = 4.0$, $Q_{22} = 4.0$, and $Q_{33} = 1.0$. The resulting sliding surfaces are:

$$\begin{aligned} \sigma_1 &= 1.0v \\ \sigma_2 &= 1.0r + 0.5(\psi - \psi_c) \end{aligned} \tag{4.10}$$

The third state is expressed explicitly as the difference between the actual and desired values. Hence, ψ_d is the Euler yaw angle in the direction of the desired navigational reference point and is measured clockwise from due north which is defined as $\psi = 0°$.

With (4.10), the control laws can be determined:

$$\begin{aligned} \delta_{br} &= 0.6642\,v + 2.2219\,r - 4.4499\,\eta^2\,satsgn(\sigma_1) \\ &\quad - 12.0713\,\eta^2\,satsgn(\sigma_2) \\ \delta_{sr} &= 0.4725\,v + 7.6752\,r - 4.6361\,\eta^2\,satsgn(\sigma_1) \\ &\quad + 13.6037\,\eta^2\,satsgn(\sigma_2) \end{aligned} \tag{4.11}$$

The resultant sliding surface pole is -0.50.

E. LINE-OF-SIGHT GUIDANCE SCHEME

1. Hit Criterion

The steering controller just designed, (4.10) and (4.11), must be able to navigate the AUV through a series of way-points in the horizontal plane. For the LOS controller, the criterion for a "hit" of a way-point has been determined to be when the AUV is within half a ship length or 8.7125 ft, i.e., the AUV has reached a way-point if it is within a circle of radius 0.5L centered on the way-point. In practice, the hit criterion used would depend upon the AUV's mission.

2. Shortest Turn

For efficiency the AUV should always turn through the shortest angle to achieve the desired heading. This is accomplished in the simulation by the excerpt of Fortran code in Figure 32. It was decided that angles within the simulation program would kept between $-\pi$ and π by using the intrinsic Fortran function ATAN2. This results in a discontinuity at due south, vice due north.

```
C
C    CALCULATE THE QUICKEST ROUTE PSID
C
     IF((PSI .GE. 0.0 .AND. PSI .LE. PIE) .AND. (PSID .GE. -PIE .AND. PSID .LE.
        PSI-PIE)) THEN
            PSID = PSID + 2.0*PIE
     ELSEIF((PSI .GE. -PIE .AND. PSI .LT. 0.0) .AND. (PSID .GT. PSI+PIE .AND. PSID
        .LE. PIE)) THEN
            PSID = PSID - 2.0*PIE
     ENDIF
```

Figure 32. Fortran Code for Calculating the Shortest Turn

ψ_d is the only exception to the rule of angles from $-\pi$ to π. The Fortran code of Figure 32 allows ψ_d to be less than $-\pi$ or greater than π in situations when ψ and ψ_d are on opposite sides of due south and the quickest route is across the discontinuity. This allows the AUV to "chase" the desired heading across the discontinuity. This is very similar to the method used by Lienard [Ref. 4].

F. SIMULATION

Programs SDVLOSRR300PSF_LQR.FOR (see Appendix A) and PLOT7A.FOR (see Appendix A) were written to simulate the AUV in the horizontal plane with perfect state feedback (PSF). Figures 33 and 34 show Run 16 conducted at an ordered speed of u = 6.0 ft/s, ϕ = 2.0, and η^2 = 0.5. Lienard's speed controller [Ref. 3] was again employed. The speed controller's saturation function was kept at ϕ = 1.0 for all runs because it was found to give the best results.

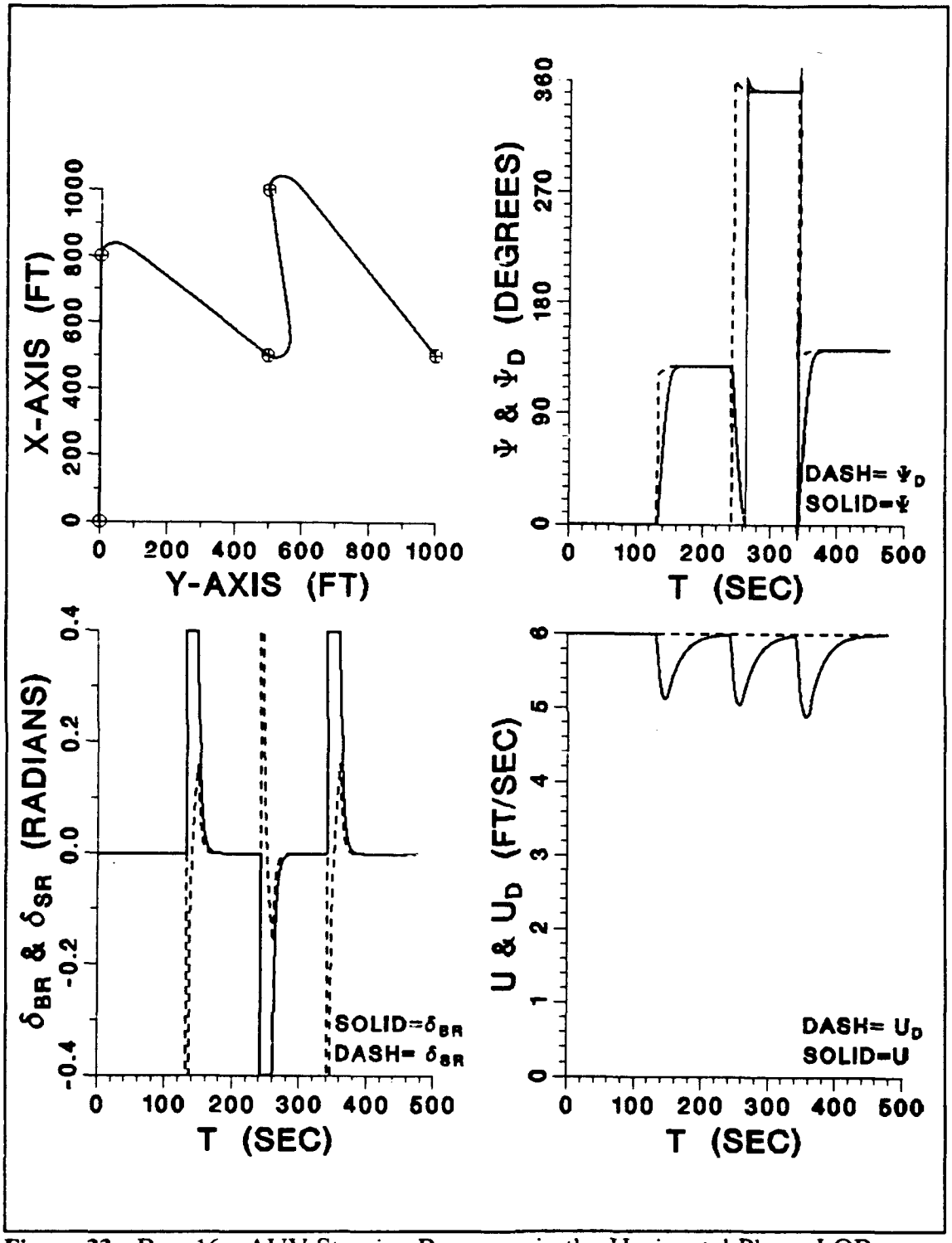


Figure 33. Run 16 - AUV Steering Response in the Horizontal Plane, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, Perfect State Feedback, No Current)

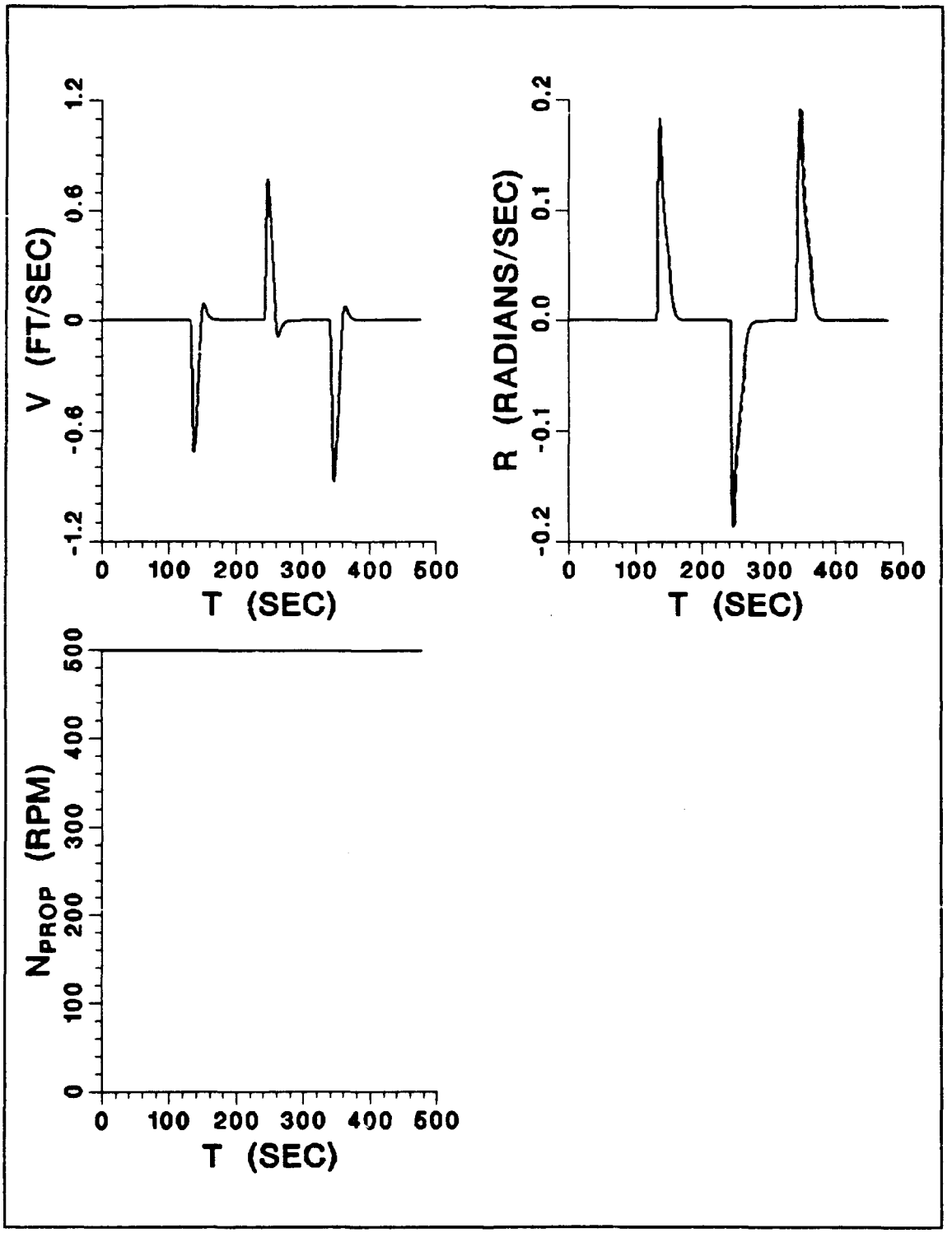


Figure 34. Run 16 - (continued)

Initial conditions of the simulation at $t = 0$ are the vehicle headed north ($\psi = 0°$) at ordered speed u_d with rudders centered. The positive x-axis is due north and the positive y-axis is due east. The small circles on the global plot of Figure 33 mark the location of the way-points. (Nothing should be inferred from the circles' size.) The global plot shows the vehicle smartly hitting the desired way-points.

The graph of rudder action deserves close scrutinization. Although the two rudders have the same effect on the vehicle, i.e., the same hydrodynamic coefficients, the control package does not order symmetrical action of the rudders. The rudders initially deflect in opposite directions as expected, but the stern rudder returns to zero quicker. In fact, the stern rudder continues past center and deflects a small amount in the same direction as the bow rudder. Thus, the stern rudder prevents overshoot. This action was not consciously designed into the controller and could not have been predicted. Each rudder operates independently of the other. This is a result of Utkin's MIMO sliding mode method.

G. STEADY-STATE DISTURBANCE COMPENSATION

1. Theory

Figures 35 and 36 show the results of Run 17 with a steady-state current present. The current is expressed as absolute velocity components in the inertial global frame, $u_{co} = 2.0$ ft/s and $v_{co} = 2.0$ ft/s. Thus, the total current has velocity of 2.828 ft/s directed towards the north-east, i.e., $\psi = 45°$. This is a significant current of almost 50% of the AUV's maximum speed.

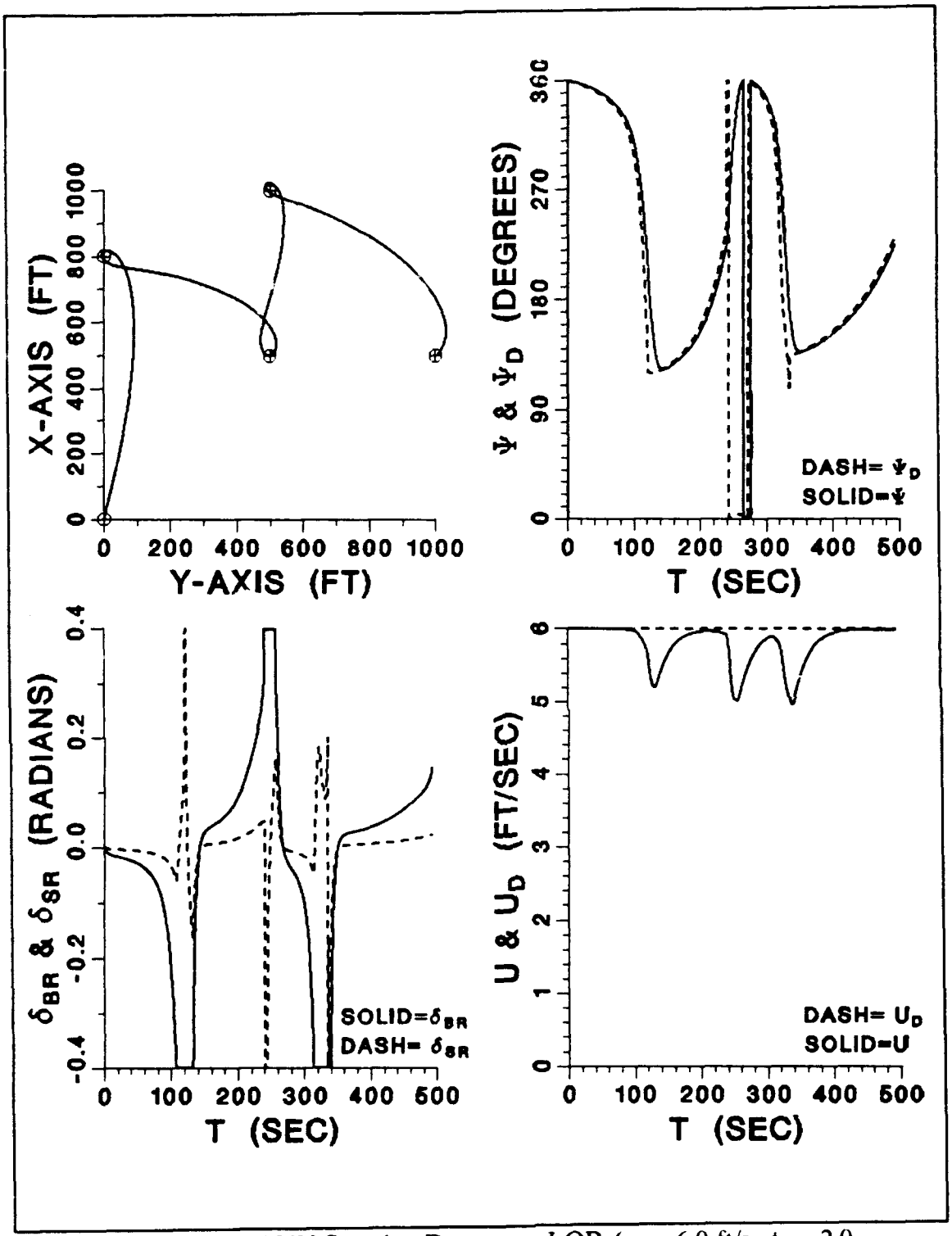


Figure 35. Run 17 - AUV Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, PSF, u_{co} = v_{co} = 2.0 ft/s, No Disturbance Compensation)

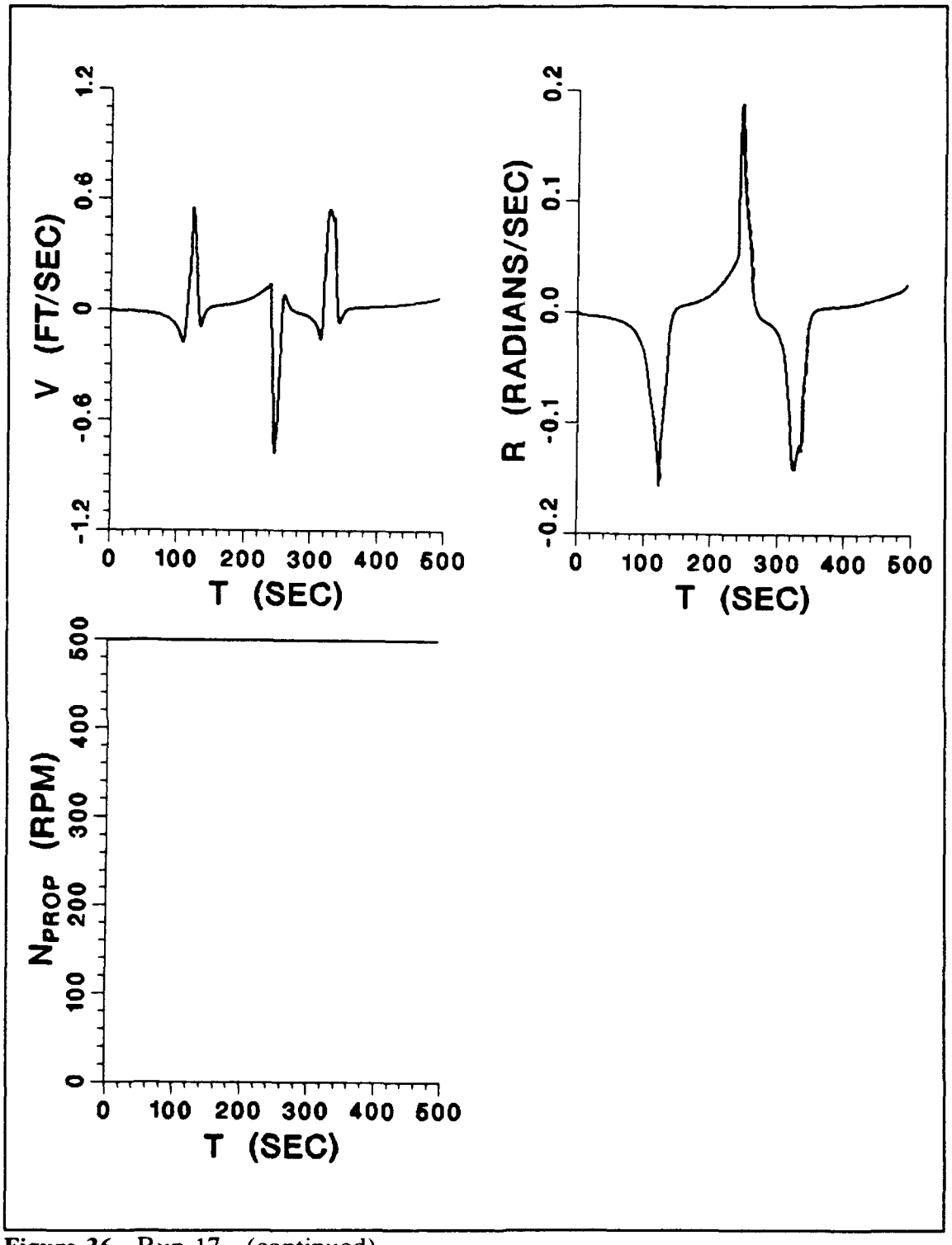


Figure 36. Run 17 - (continued)

The global plot on Figure 35 exemplifies the problem of uncompensated disturbances. The vehicle hits the way-points, but does so via a circuitous route. Also, in contrast to common sense the rudder angles are initially small and utilize their full strength only during the final stage of the approach to the way-point. Let's determine the nature of the problem and investigate a way to compensate.

First, the linear state space system of (4.8) can be expressed in general terms:

$$\begin{aligned} \dot{v} &= a_{11}v + a_{12}r + b_{11}\delta_{br} + b_{12}\delta_{sr} \\ \dot{r} &= a_{21}v + a_{22}r + b_{21}\delta_{br} + b_{22}\delta_{sr} \\ \dot{\psi} &= r \end{aligned} \tag{4.12}$$

In a like manner, the controller equations expressed in general are:

$$\begin{aligned} \sigma_1 &= s_{11}v + s_{12}r + s_{13}(\psi - \psi_d) \\ \sigma_2 &= s_{21}v + s_{22}r + s_{23}(\psi - \psi_d) \end{aligned} \tag{4.13}$$

$$\begin{aligned} \delta_{br} &= k_{11}v + k_{12}r + k_{13}satsgn(\sigma_1) + k_{14}satsgn(\sigma_2) \\ \delta_{sr} &= k_{21}v + k_{22}r + k_{23}satsgn(\sigma_1) + k_{24}satsgn(\sigma_2) \end{aligned} \tag{4.14}$$

where the η^2 terms have been incorporated within the gains k_{13}, k_{14}, k_{23}, and k_{24} in (4.14). With current present, the inertial position rates (4.7) become:

$$\begin{aligned} \dot{x} &= u_{co} + u\cos\psi - v\sin\psi \\ \dot{y} &= v_{co} + u\sin\psi + v\cos\psi \end{aligned} \tag{4.15}$$

Equation (4.15) can also be expressed in the local reference frame formed by the leg of two consecutive way-points:

$$\begin{aligned} \dot{x}' &= u_c + u\cos\psi' - v\sin\psi' \\ \dot{y}' &= v_c + u\sin\psi' + v\cos\psi' \end{aligned} \tag{4.16}$$

where u_c and v_c are the current components with respect to the local leg, and ψ' is the heading angle with respect to the local leg:

$$\psi' \equiv (\psi - \alpha) \tag{4.17}$$

$$\alpha \equiv \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{4.18}$$

The variable α in (4.17) and (4.18) is the angle the local leg makes with the x-axis which is due north or $\psi = 0°$. Positive α is defined as clockwise. The points (x_1, y_1) and (x_2, y_2) are the coordinates of the two way-points that define the local leg. Finally, the local components of current are related to the global components by:

$$\begin{aligned} u_c &= v_{co}\sin\alpha + u_{co}\cos\alpha \\ v_c &= v_{co}\cos\alpha - u_{co}\sin\alpha \end{aligned} \tag{4.19}$$

When the AUV is at a steady-state condition, (4.12) becomes:

$$\begin{aligned} \dot{v} &= a_{11}v + a_{12}r + b_{11}\delta_{br} + b_{12}\delta_{sr} = 0 \\ \dot{r} &= a_{21}v + a_{22}r + b_{21}\delta_{br} + b_{22}\delta_{sr} = 0 \\ \dot{\psi} &= r = 0 \end{aligned} \tag{4.20}$$

Assuming zero deflection of the rudders at steady-state even in the presence of a current, (4.20) results in:

$$v = r = \delta_{br} = \delta_{sr} = 0 \tag{4.21}$$

and (4.21) then leads to the conclusion that:

$$\left.\begin{aligned} \sigma_1 &= 0 \\ \sigma_2 &= 0 \end{aligned}\right\} \rightarrow \psi' = \psi'_d \rightarrow \psi = \psi_d \tag{4.22}$$

Hence, the AUV achieves desired heading even in the presence of a current! In that case, however, $\dot{y}'$ is nonzero and y' increases linearly with time. The result is the AUV is continuously pushed laterally away from the leg by the v_c component of current.

What is the steady-state value of ψ'_d? Through the LOS guidance scheme, the desired heading becomes a function of the instantaneous ship position and the next desired way-point. The local desired heading ψ'_d at steady-state can be computed from the kinematics:

$$\begin{aligned} \dot{x}' &= u_c + u\cos\psi_d' \\ \dot{y}' &= v_c + u\sin\psi_d' \end{aligned} \tag{4.23}$$

and from the LOS requirement:

$$\tan\psi_d' = \frac{y'}{x'} \tag{4.24}$$

Using (4.23) and (4.24), we can see that:

$$\tan\psi_d' = \frac{v_c}{u_c} \tag{4.25}$$

Thus, the LOS guidance law eventually aligns the AUV with the current when it approaches a way-point! This action can be seen in the global plot of Figure 35.

The requirement $\dot{y}' = 0$ can be achieved by modifying the sliding surfaces to:

$$\begin{aligned} \sigma_1 &= s_{11}v + s_{12}r + s_{13}(\psi - \psi_d) + s_{13}\arcsin\left(\frac{v_c}{u}\right) \\ \sigma_2 &= s_{21}v + s_{22}r + s_{23}(\psi - \psi_d) + s_{23}\arcsin\left(\frac{v_c}{u}\right) \end{aligned} \tag{4.26}$$

Then at steady-state (4.22) ($\sigma_1 = \sigma_2 = 0$) and hence (4.21) are achieved, but there is a heading error at steady-state of:

$$\psi'_{error} = \psi_{error} \simeq -\arcsin\left(\frac{v_c}{u}\right) \tag{4.27}$$

but in this case the desired result of $y' = 0$ is achieved. To put it in simple terms, the AUV overcomes the local lateral component of current by keeping its bow headed into the current an amount ψ_{error} in (4.27).

The sliding surfaces for the AUV's LOS controller can now be explicitly expressed:

$$\begin{aligned} \sigma_1 &= 1.0\,v \\ \sigma_2 &= 1.0\,r + 0.5\,(\psi - \psi_d) + 0.5\arcsin\left(\frac{v_c}{u}\right) \end{aligned} \tag{4.28}$$

The reader should note that we could have assumed a zero local heading error ($\psi' = 0$) at steady-state, vice the assumptions of (4.21). This would lead to nonzero rudder deflections at steady-state. We'll explore this different approach in Chapter V for the CTE guidance steering controller.

2. AUV Simulation with Steady-State Disturbance Compensation

Fortran program SDVLOSRR300PSF_LQR.FOR was modified to enact the disturbance compensation of (4.28). Figures 37 and 38 are for Run 18 with the modified sliding surfaces. The only difference between Run 18 and Run 17 is disturbance compensation, so they can be easily compared. Disturbance compensation clearly corrects for the presence of the current. The AUV overcomes

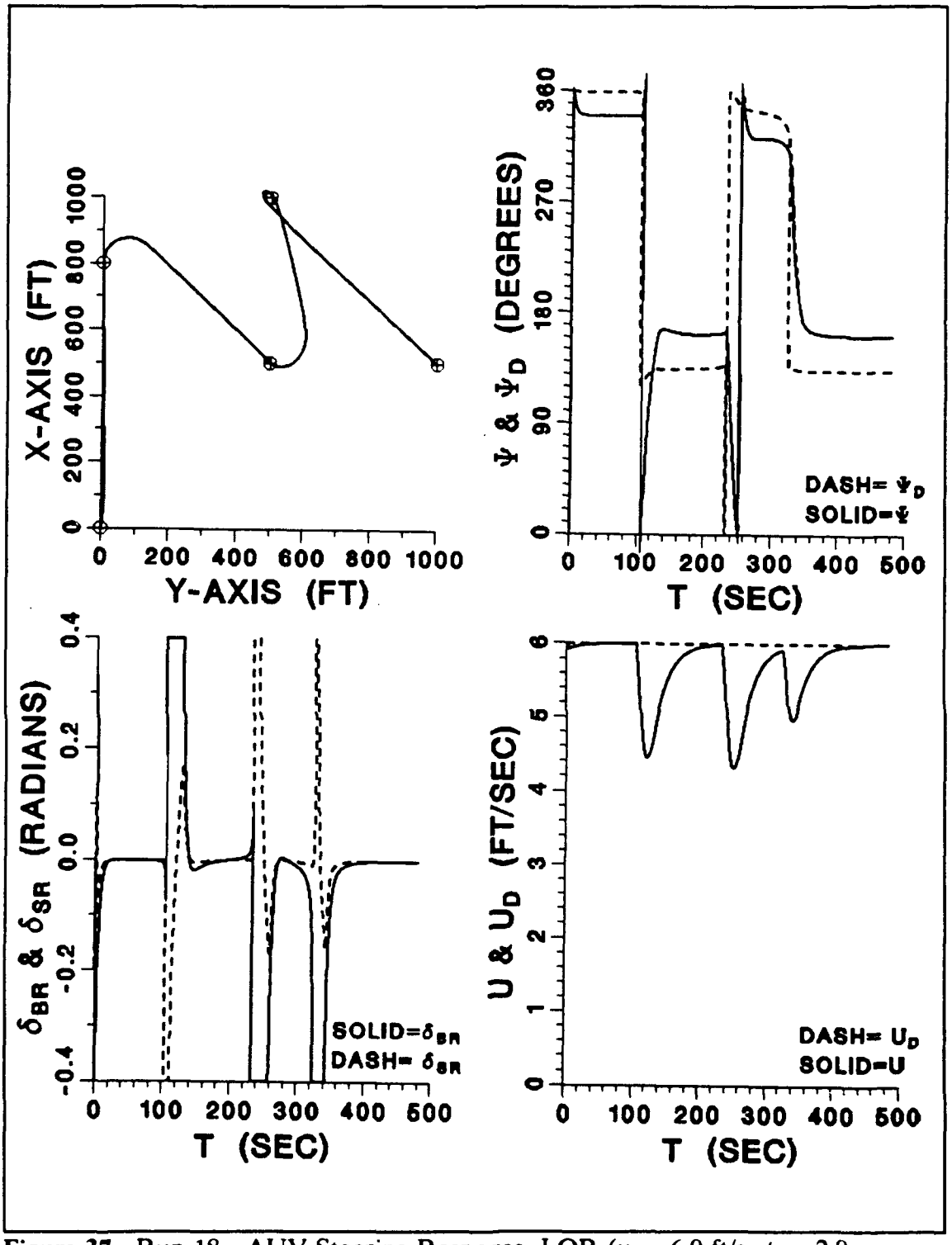


Figure 37. Run 18 - AUV Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, PSF, $u_{co} = v_{co}$ = 2.0 ft/s, Disturbance Compensation)

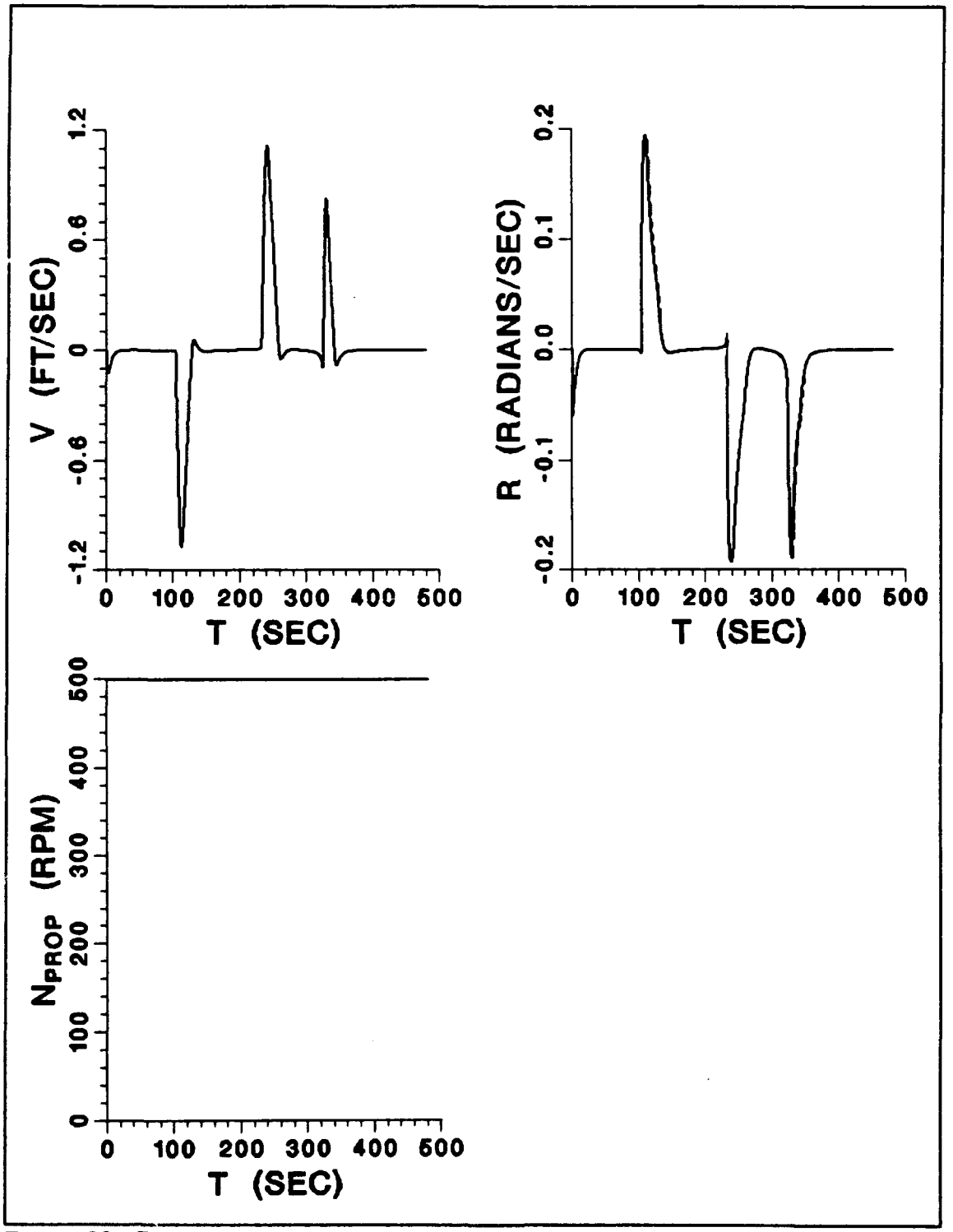


Figure 38. Run 18 - (continued)

the current and hits the designated way-points. It should be emphasized that in all simulations where current is present, the full strength of the current is instantly applied at t = 0, thus creating a severe start-up test for the controller.

Unfortunately, one difficulty remains. Ocean current is not directly measurable by the AUV. Additionally, at the time of this thesis it was not clear if the NPS AUV would be able to measure sway velocity v. Assuming the worst, v will also be unmeasured. As in the diving situation, observers must be designed for state estimation.

H. OBSERVERS

1. Design

A reduced order observer can be designed based on x, y, ψ, and r measurements to estimate the sway velocity v and the local current components u_c and v_c. The current component u_c does not need to be observed because the control package does not use it, but it will be included in the output of the reduced order observer as an exercise.

The procedure is the same as in the diving controller; however, to start the design process the state vector is augmented. The linear state space representation becomes:

$$
\begin{aligned}
\dot{v} &= a_{11}uv + a_{12}ur + b_{11}u^2\delta_{br} + b_{12}u^2\delta_{sr} \\
\dot{r} &= a_{21}uv + a_{22}ur + b_{21}u^2\delta_{br} + b_{22}u^2\delta_{sr} \\
\dot{\psi}' &= r \\
\dot{y}' &= v_c + v + u\psi' \\
\dot{x}' &= u_c + u \equiv U_c \\
\dot{v}_c &= 0 \\
\dot{U}_c &= 0
\end{aligned}
\tag{4.29}
$$

The reduced order observer equations become:

$$\hat{v} = \hat{Z}_1 + L_{12}r \tag{4.30}$$

$$\hat{v}_c = \hat{Z}_2 + L_{23}y' \tag{4.31}$$

$$\hat{a}_c = \hat{Z}_3 + L_{34}x' - u\cos\psi' \tag{4.32}$$

$$
\begin{aligned}
\dot{\hat{Z}}_1 = S_1\hat{Z}_1 + (a_{12}u - L_{12}a_{22}u + S_1L_{12})r + (b_{11} - b_{21}L_{12})u^2\delta_{br} \\
+ (b_{12} - b_{22}L_{12})u^2\delta_{sr}
\end{aligned}
\tag{4.33}
$$

$$\dot{\hat{Z}}_2 = S_2Z_1 + S_2\hat{Z}_2 - L_{23}u\sin\psi' + S_2L_{23}y' + S_2L_{12}r \tag{4.34}$$

$$\dot{\hat{Z}}_3 = S_3\hat{Z}_3 + S_3L_{34}x' \tag{4.35}$$

$$L_{12} = \frac{a_{11}u - S_1}{a_{21}u} \qquad L_{23} = -S_2 \qquad L_{34} = -S_3 \tag{4.36}$$

$$S_1,\ S_2,\ S_3 \equiv \textit{observer poles} \tag{4.37}$$

The observer poles were selected at -1.0, -1.1, and -1.2. The estimated values $\hat{a}_c$ and $\hat{v}_c$ are updated every time a new straight line segment is encountered. The reduces the transients in the observer during transition between one way-point to the next.

2. Simulation with Observers

Programs SDVLOSRR300OBS_LQR.FOR (see Appendix D) and PLOT8.FOR (see Appendix A) simulate the AUV with disturbance compensation and observers. Figures 39 and 40 show Run 19 which is a duplicate of the previous steering simulation runs but with observers. The observers do a very good job of estimating the unknown states, and vehicle control is not degraded by using these estimates.

All previous steering simulations runs (Runs 16-19) have been executed with a simulation time step of $\Delta t = 0.01$ s. The control packages were also updated every 0.01 s. In reality, the AUV's controller may not be updated this often. To investigate what may happen, program SDVLOSRR300OBS_LQR.FOR was modified so the control package was only updated ever 0.1 s and Figures 41 and 42 show the results. The only effect on the vehicle is a slight degradation of the estimates of the unknown states. This degradation is not significant enough to jeopardize vehicle control even slightly.

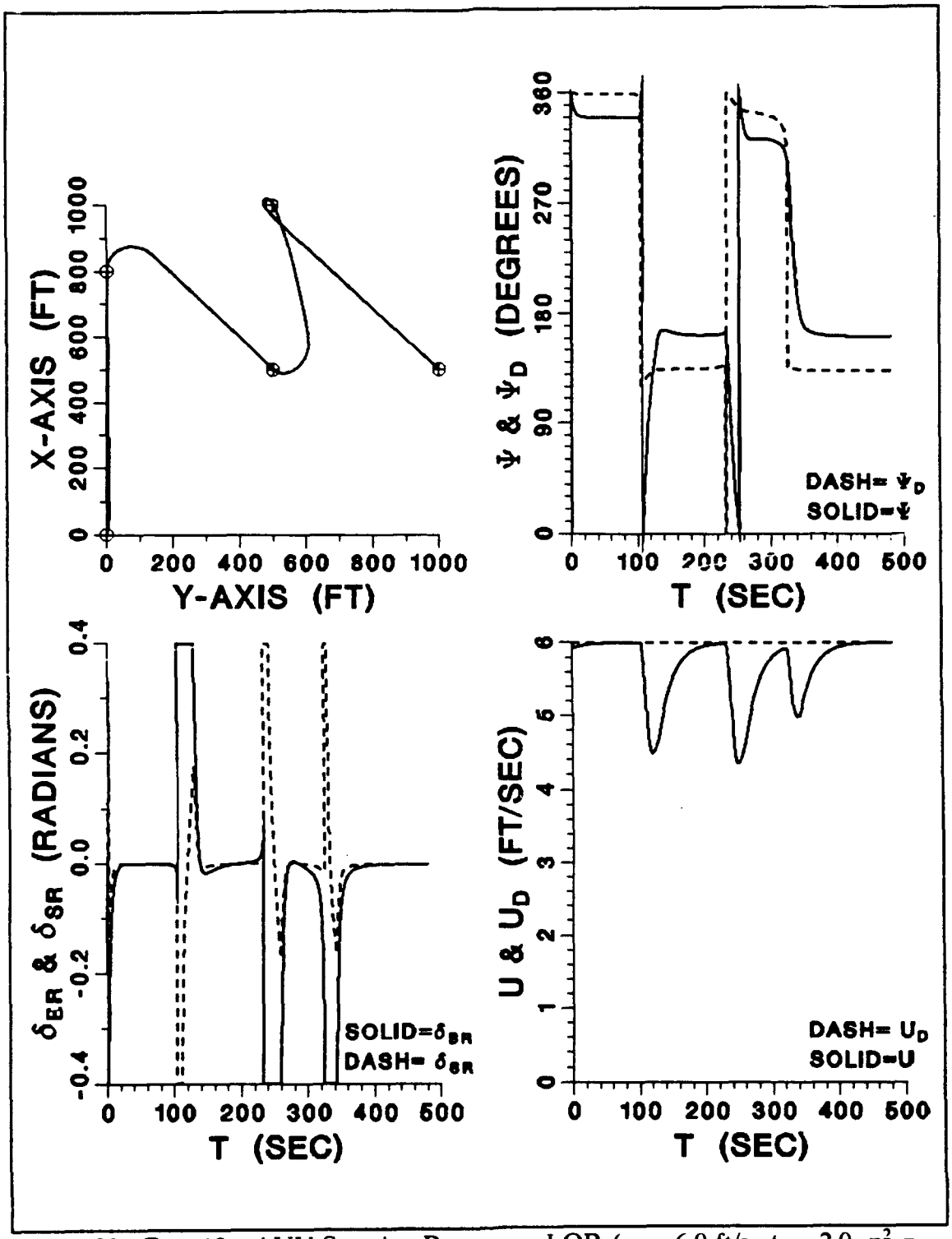


Figure 39. Run 19 - AUV Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, Observers, $u_{co} = v_{co}$ = 2.0 ft/s, Disturbance Compensation)

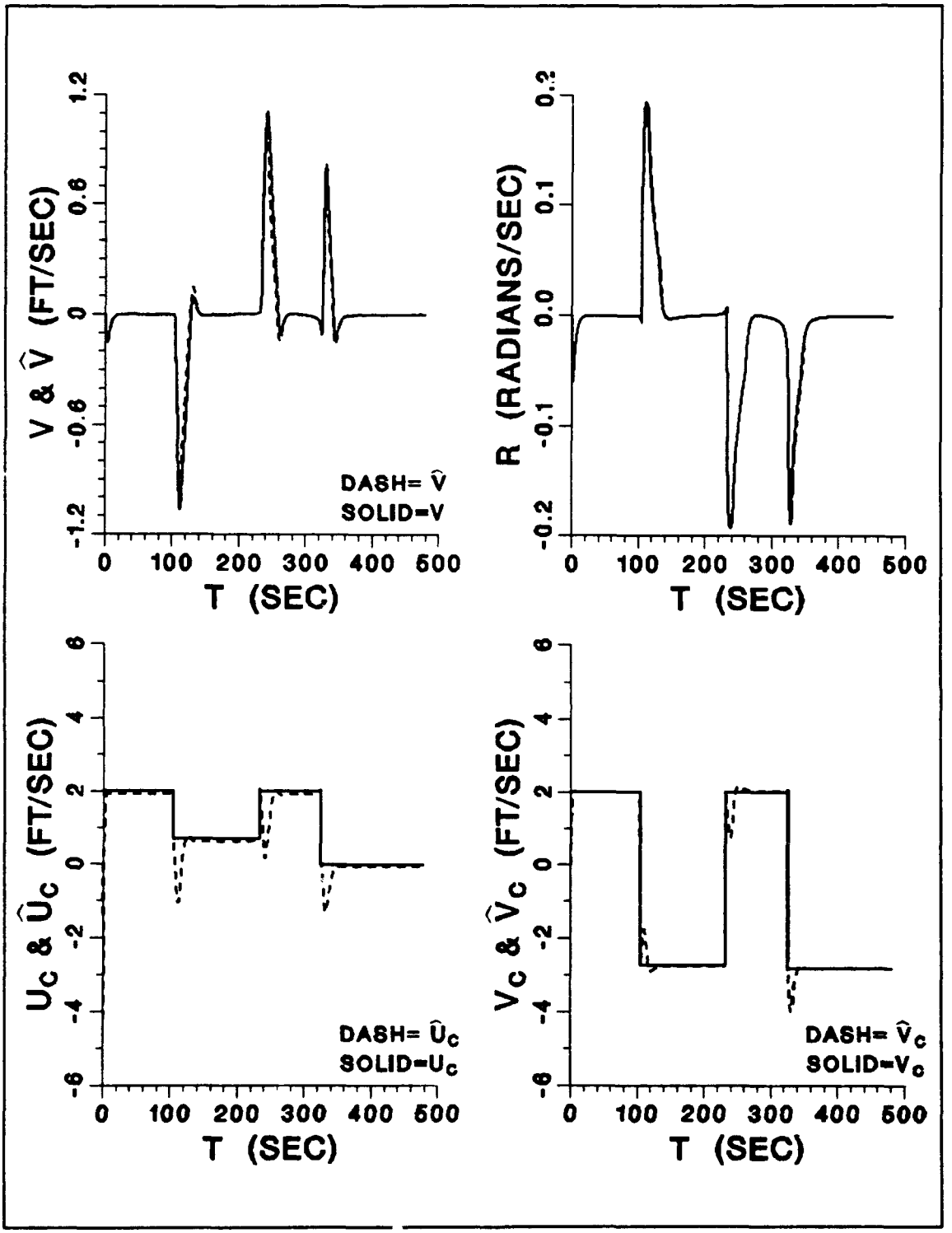


Figure 40. Run 19 - (continued)

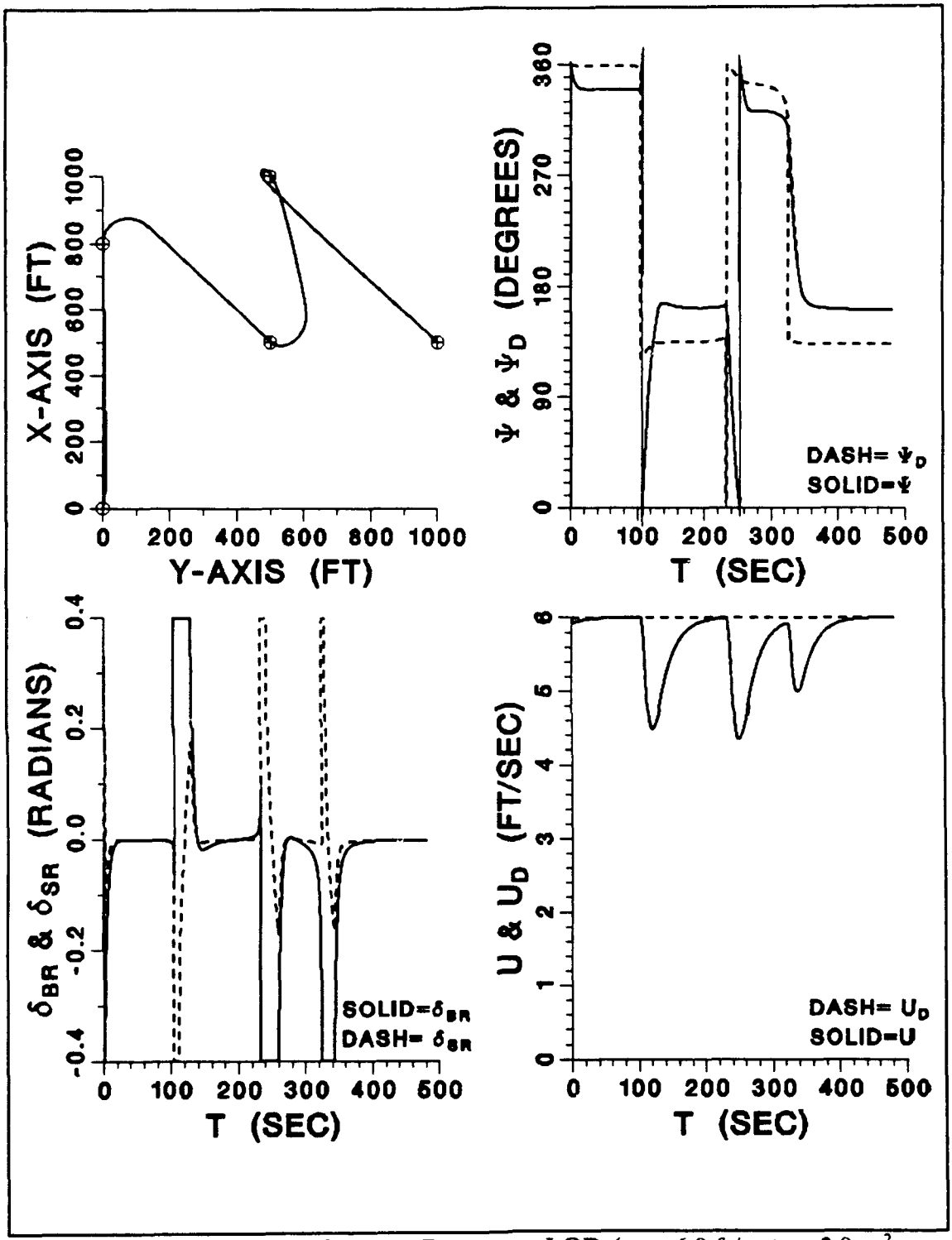


Figure 41. Run 20 - AUV Steering Response, LQR (u = 6.0 ft/s, ϕ = 2.0, η^2 = 0.5, Observers, u_{co} = v_{co} = 2.0 ft/s, Disturbance Compensation)

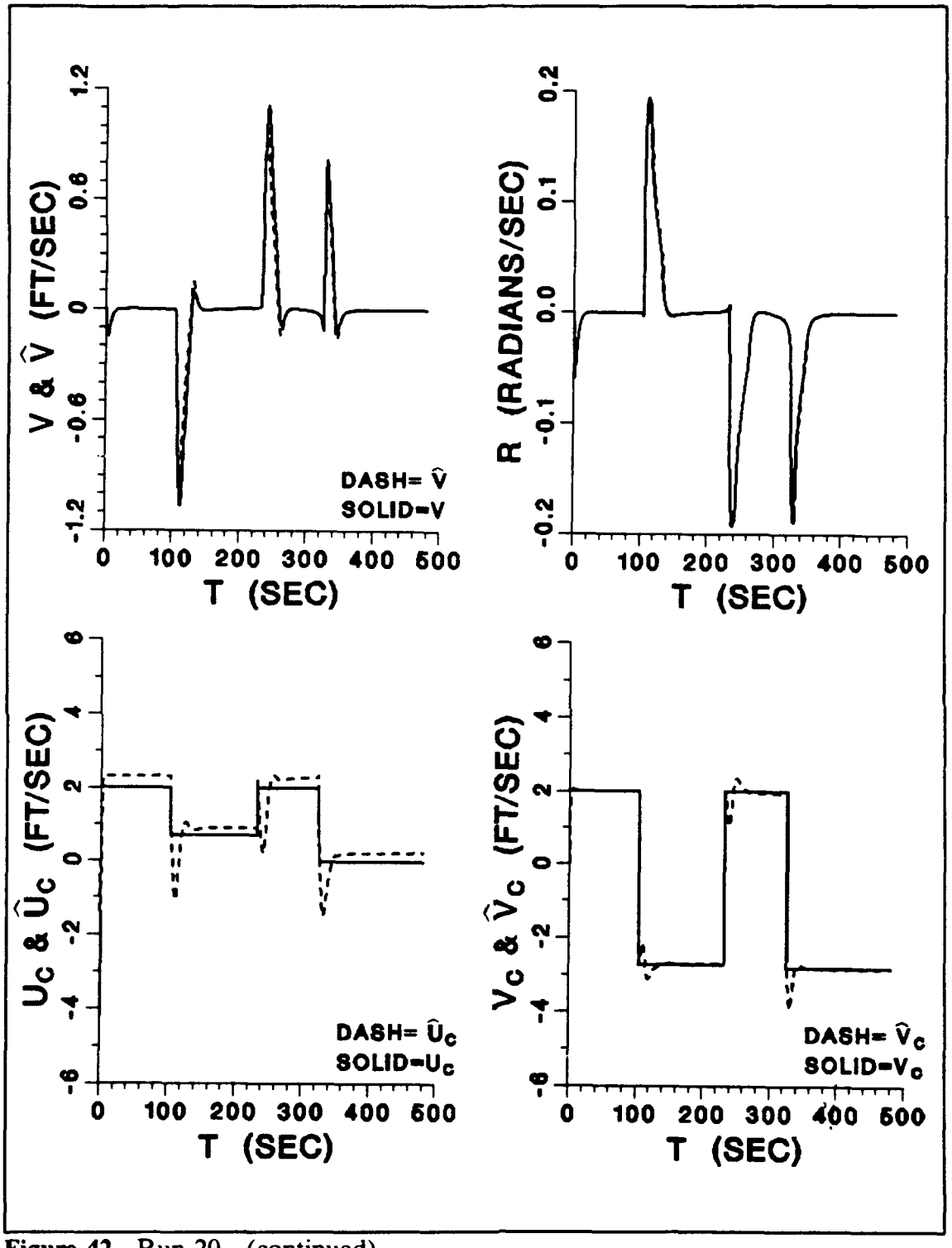


Figure 42. Run 20 - (continued)

I. CONCLUDING REMARKS

A MIMO sliding mode controller has been designed for the AUV which utilized a line-of-sight guidance scheme. The controller proved to be very effective. Disturbance compensation was achieved by means of a feedforward term in the sliding surface equations. Disturbance estimation was possible by using a standard Leunberger observer. The resulting scheme demonstrated excellent path keeping characteristics in the presence of strong lateral currents wit. out any compromise in stability and robustness properties.

Line-of-sight guidance is a good navigation scheme for open ocean cruising. However, LOS may not provide adequate guidance in restricted waters because it allows the vehicle to overshoot way-points. This problem leads us to explore cross-track-error guidance in Chapter V.

V. CROSS-TRACK-ERROR GUIDANCE OF THE AUV USING MIMO SLIDING MODE CONTROL

A. INTRODUCTION

Analysis of the simulation runs in Chapter IV show that the AUV with LOS guidance is unable to follow a perfectly straight path in the presence of a steady-state disturbance, even with compensation. This is adequate for open ocean cruising; however, LOS guidance will not fulfill the tolerance requirements of operations in restricted waters unless many closely spaced way-points are programmed into the AUV's mission planning computer. Cross-Track-Error (CTE) guidance is an alternate scheme for steering control that will enable the AUV to follow a straight path.

B. CROSS-TRACK-ERROR GUIDANCE SCHEME

All the tools we need to design a CTE guidance steering controller were introduced in Chapter IV. The governing equations, (4.2) to (4.7), still apply. Also valid are the equations expressed in the local reference frame, (4.16) to (4.19). When the AUV reaches a new way-point, that way-point becomes the new origin of a local coordinate system formed with the next ordered way-point. Obviously this scheme requires the augmentation of the state vector used for LOS guidance with the cross-track-error variable, y'. The new linear state space system used for controller design can be expressed in general terms without disturbances as:

$$
\begin{aligned}
\dot{v} &= a_{11}v + a_{12}r + b_{11}\delta_{br} + b_{12}\delta_{sr} \\
\dot{r} &= a_{21}v + a_{22}r + b_{21}\delta_{br} + b_{22}\delta_{sr} \\
\dot{\psi}' &= r \\
\dot{y}' &= v + a_{43}\psi'
\end{aligned}
\tag{5.1}
$$

The constants in (5.1) have been determined by linearization at a particular surge velocity, u. As in the previous chapter, a surge velocity of u = 3.00 ft/s has been chosen to produce the linearization (5.1).

C. CONTROLLER DESIGN BY LQR TECHNIQUE

The LQR method utilized with $Q_{11} = Q_{22} = Q_{33} = 100$ and $Q_{44} = 1$ results in a control package with the following equations:

$$
\begin{aligned}
\sigma_1 &= 1.0000v + 0.0920\psi' + 0.0393y' \\
\sigma_2 &= 1.0000r + 1.2423\psi' + 0.0920y'
\end{aligned}
\tag{5.2}
$$

$$
\begin{aligned}
\delta_{br} &= -0.6207v - 7.1479r - 3.8547\psi' - 4.4499\eta^2 satsgn(\sigma_1) \\
&\qquad - 12.0714\eta^2 satsgn(\sigma_2) \\
\delta_{sr} &= 1.5414v + 17.3469r + 3.2066\psi' - 4.6361\eta^2 satsgn(\sigma_1) \\
&\qquad + 13.6038\eta^2 satsgn(\sigma_2)
\end{aligned}
\tag{5.3}
$$

This control package has poles of -0.3336 and -0.9480 on the sliding surfaces. The next step is to design disturbance compensation into the controller.

D. STEADY-STATE DISTURBANCE COMPENSATION

Two methods of compensating for steady underwater currents will be discussed. The first method compensates for disturbances by utilization of a heading error. This was the method used in Chapter IV with LOS guidance where the vehicle angles its bow into the current to overcome it. The second method to be discussed uses the vehicle's rudders to overcome the current. This allows the vehicle's bow to remain pointed down the track. This method exploits the unique capabilities of the MIMO CTE steering controller.

1. Theory of Heading Error Compensation

To simplify the analysis, the sliding surfaces (5.2) and the control laws (5.3) are expressed in general as:

$$\begin{aligned} \sigma_1 &= s_{11}v + s_{12}r + s_{13}\psi' + s_{14}y' \\ \sigma_2 &= s_{21}v + s_{22}r + s_{23}\psi' + s_{24}y' \end{aligned} \tag{5.4}$$

$$\begin{aligned} \delta_{br} &= k_{11}v + k_{12}r + k_{13}\psi' + k_{14}y' + k_{15}\eta^2 satsgn(\sigma_1) \\ &\quad + k_{16}\eta^2 satsgn(\sigma_2) \\ \delta_{sr} &= k_{21}v + k_{22}r + k_{23}\psi' + k_{24}y' + k_{25}\eta^2 satsgn(\sigma_1) \\ &\quad + k_{26}\eta^2 satsgn(\sigma_2) \end{aligned} \tag{5.5}$$

The reader should note that $k_{14} = k_{24} = 0$, but the equations will be left in this very general form as a function of the entire state vector.

As stated previously, heading error compensation is identical with the procedure used in Chapter IV for the LOS controller. The reader should note that the following analysis is conducted at steady-state conditions. The procedure starts with steady-state assumptions (4.21) repeated here for convenience:

$$v = r = \delta_{br} = \delta_{sr} = 0 \tag{5.6}$$

Equation (5.6) leads to a nonzero heading error at steady-state, or in other words the AUV fights the disturbance by angling its bow into the current, not by using its rudders.

The analysis to compensate for disturbances is complicated by the additional state y'. The requirement of realizing a CTE guidance scheme results in the CTE part of (4.16) becoming:

$$\dot{y}' = v_c + u \sin\psi' = 0 \tag{5.7}$$

Solving (5.7), we get the resultant steady-state heading error:

$$\psi' = -\arcsin\left(\frac{v_c}{u}\right) \tag{5.8}$$

Now we must see if this heading error has any undesired effects on the controller. Equations (5.6), (5.7), and (5.8) result in the control laws (5.5) simplifying to:

$$k_{15}\eta^2 satsgn(\sigma_1) + k_{16}\eta^2 satsgn(\sigma_2) = k_{13}\arcsin\left(\frac{v_c}{u}\right)$$
$$k_{25}\eta^2 satsgn(\sigma_1) + k_{26}\eta^2 satsgn(\sigma_2) = k_{23}\arcsin\left(\frac{v_c}{u}\right) \tag{5.9}$$

at steady-state. Solving (5.9) for the two saturation functions:

$$satsgn(\sigma_1) = \frac{1}{\eta^2}\left(\frac{k_{13}k_{26} - k_{16}k_{23}}{k_{15}k_{26} - k_{16}k_{25}}\right)\arcsin\left(\frac{v_c}{u}\right)$$
$$satsgn(\sigma_2) = \frac{1}{\eta^2}\left(\frac{k_{15}k_{23} - k_{13}k_{25}}{k_{15}k_{26} - k_{16}k_{25}}\right)\arcsin\left(\frac{v_c}{u}\right) \tag{5.10}$$

For purposes of controller stability, at steady-state we desire the saturation functions to be:

$$satsgn(\sigma_1) \le 1$$
$$satsgn(\sigma_2) \le 1 \tag{5.11}$$

which establishes the lower limit on η^2. Then it follows from (5.11):

$$satsgn(\sigma_1) = \frac{\sigma_1}{\phi}$$
$$satsgn(\sigma_2) = \frac{\sigma_2}{\phi} \tag{5.12}$$

Combining (5.10) and (5.12), we can solve for the uncompensated sliding surfaces at steady-state in the presence of a constant disturbance:

$$\sigma_1 = \frac{\phi}{\eta^2}\left(\frac{k_{13}k_{26} - k_{16}k_{23}}{k_{15}k_{26} - k_{16}k_{25}}\right)\arcsin\left(\frac{v_c}{u}\right)$$
$$\sigma_2 = \frac{\phi}{\eta^2}\left(\frac{k_{15}k_{23} - k_{13}k_{25}}{k_{15}k_{26} - k_{16}k_{25}}\right)\arcsin\left(\frac{v_c}{u}\right) \tag{5.13}$$

Back substitution of (5.13) into the sliding surface equations (5.4) yields two expressions for the uncompensated steady-state cross-track error:

$$y' = \frac{1}{s_{14}}\left[\frac{\phi}{\eta^2}\left(\frac{k_{13}k_{26} - k_{16}k_{23}}{k_{15}k_{26} - k_{16}k_{25}}\right) + s_{13}\right]\arcsin\left(\frac{v_c}{u}\right)$$
$$y' = \frac{1}{s_{24}}\left[\frac{\phi}{\eta^2}\left(\frac{k_{15}k_{23} - k_{13}k_{25}}{k_{15}k_{26} - k_{16}k_{25}}\right) + s_{23}\right]\arcsin\left(\frac{v_c}{u}\right) \tag{5.14}$$

Effective disturbance compensation can be achieved by eliminating the effect of these two terms. To eliminate (5.13) and (5.14) the sliding surfaces are augmented with the additional terms:

$$\sigma_1 = s_{11}v + s_{12}r + s_{13}\psi' + s_{14}y' + \left[\frac{\phi}{\eta^2}\left(\frac{k_{13}k_{26} - k_{16}k_{23}}{k_{15}k_{26} - k_{16}k_{25}}\right) + s_{13}\right]\arcsin\left(\frac{v_c}{u}\right)$$
$$\sigma_2 = s_{21}v + s_{22}r + s_{23}\psi' + s_{24}y' + \left[\frac{\phi}{\eta^2}\left(\frac{k_{15}k_{23} - k_{13}k_{25}}{k_{15}k_{26} - k_{16}k_{25}}\right) + s_{23}\right]\arcsin\left(\frac{v_c}{u}\right) \tag{5.15}$$

2. Simulation with Heading Error Compensation

The observers for v, v_c, and u_c designed in Chapter IV can be used for the CTE controller with no modification. Programs SDVCTERR300OBS_LQR.FOR (see Appendix E) and PLOT10.FOR (see Appendix A) were used for AUV simulation with CTE guidance. Run 21, shown in Figures 43 and 44, was conducted with a hit distance of 0.5L as was used with the LOS controller. Global currents of u_{co} = 2.0 ft/s and v_{co} = -2.0 ft/s operated on the vehicle starting at t = 0. Vehicle speed was u = 6.0 ft/s and controller parameters were ϕ = 2.0 and η^2 = 0.35.

The AUV easily follows the straight path between successive way-points even in the presence of a significant current, but the small hit criterion of 0.5L causes the AUV to overshoot the next leg in cases where the next leg must be reached through a sharp angle. The worst case can be seen in the global plot in Figure 43 at way-point (400,400). The AUV overshoots the way-point and must reverse course to fight the current back to the next leg. This wastes time and energy.

A simple but very effective adjustment is to change the hit criterion to a larger value. Run 22 is a duplicate of Run 21 but with a hit criterion of seven ship lengths or 7L. Figures 45 and 46 show the simulation results. The new hit criterion yields a vast improvement in the AUV's ability to track the path. The AUV is also able to travel farther because its path keeping has been optimized.

Increasing the hit distance does have a drawback. When the AUV must turn through a shallow angle for the next leg, 7L is too large a hit distance and the vehicle heads for the next leg too early. Hence, increasing the hit distance is a

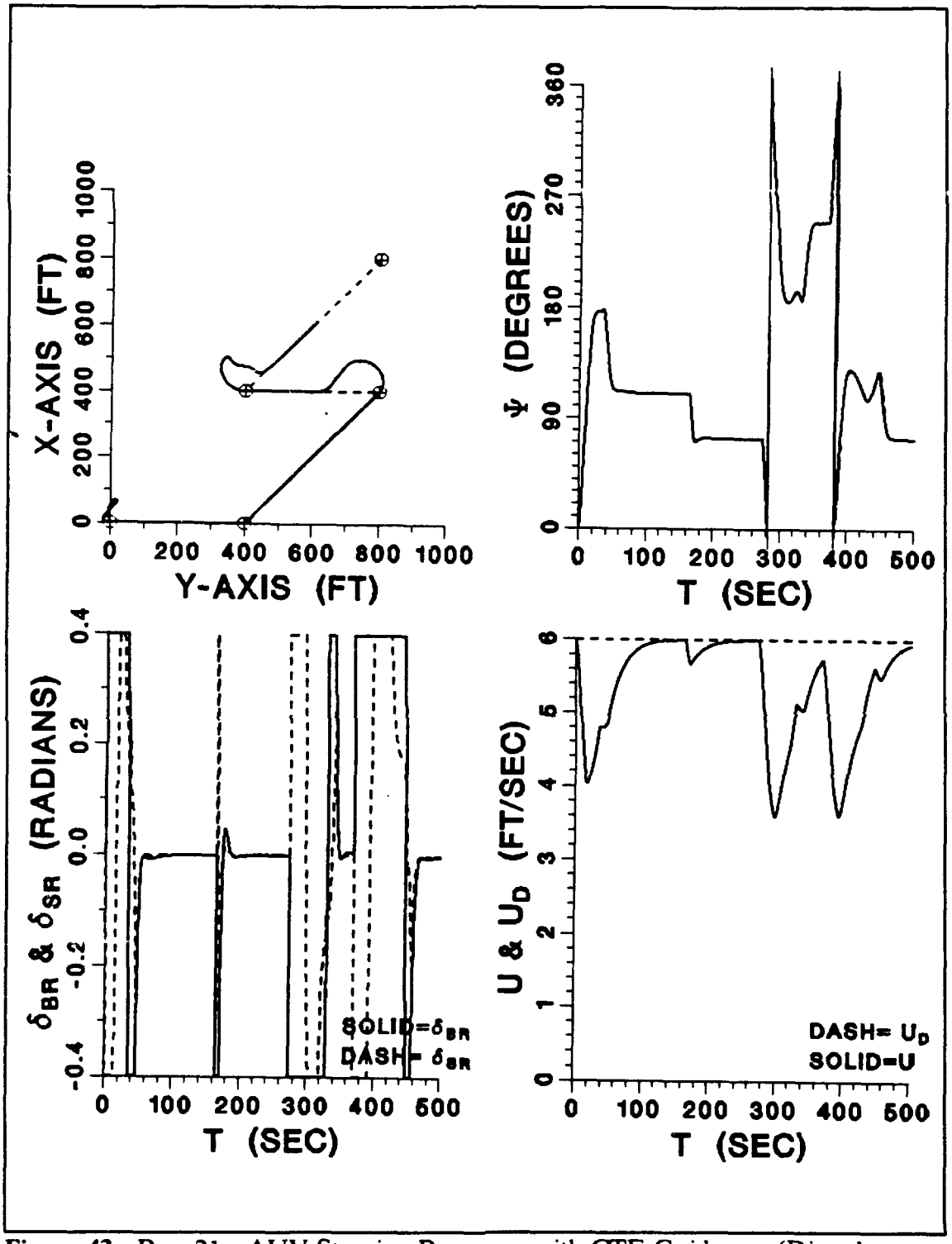


Figure 43. Run 21 - AUV Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, Hit = 0.5L)

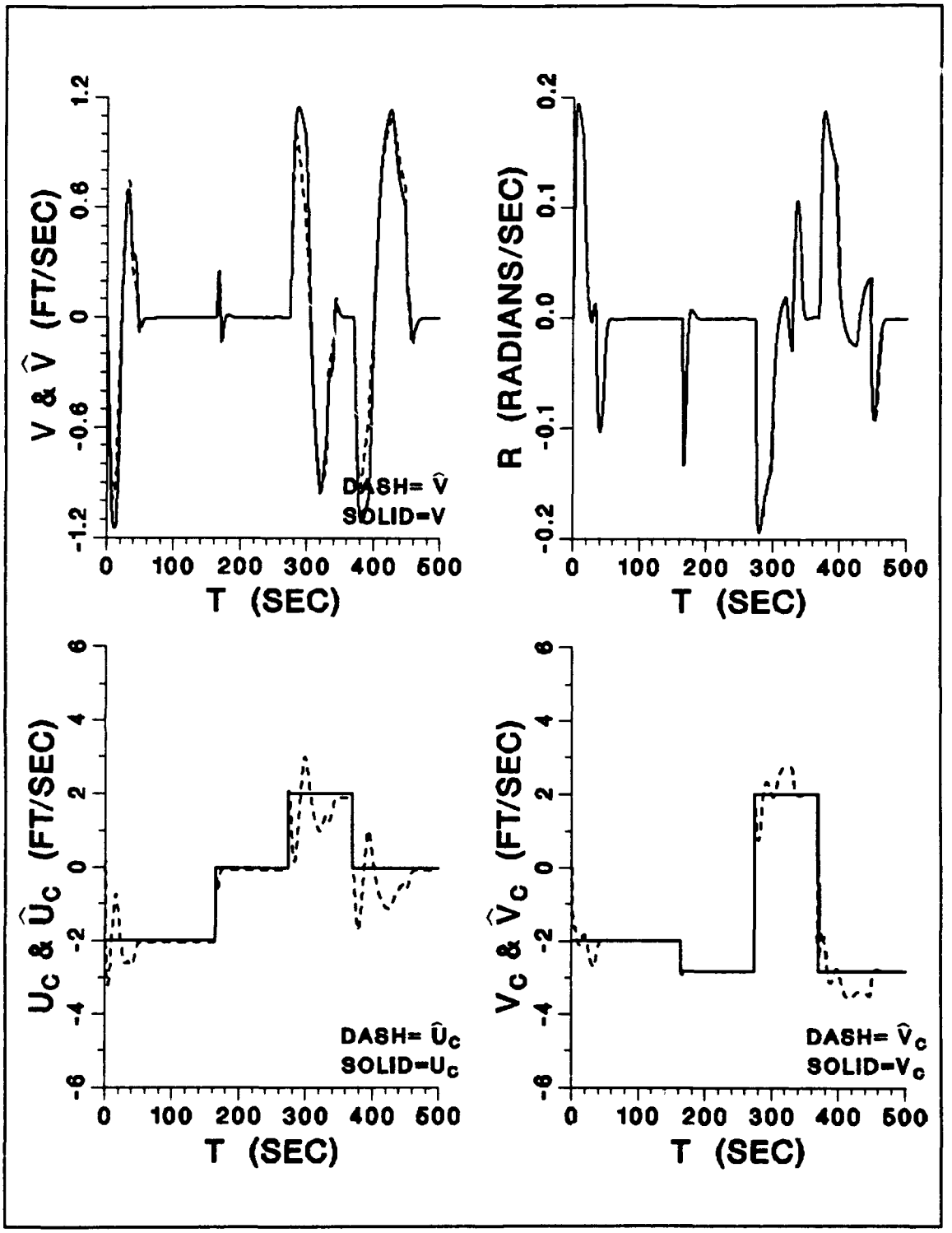


Figure 44. Run 21 - (continued)

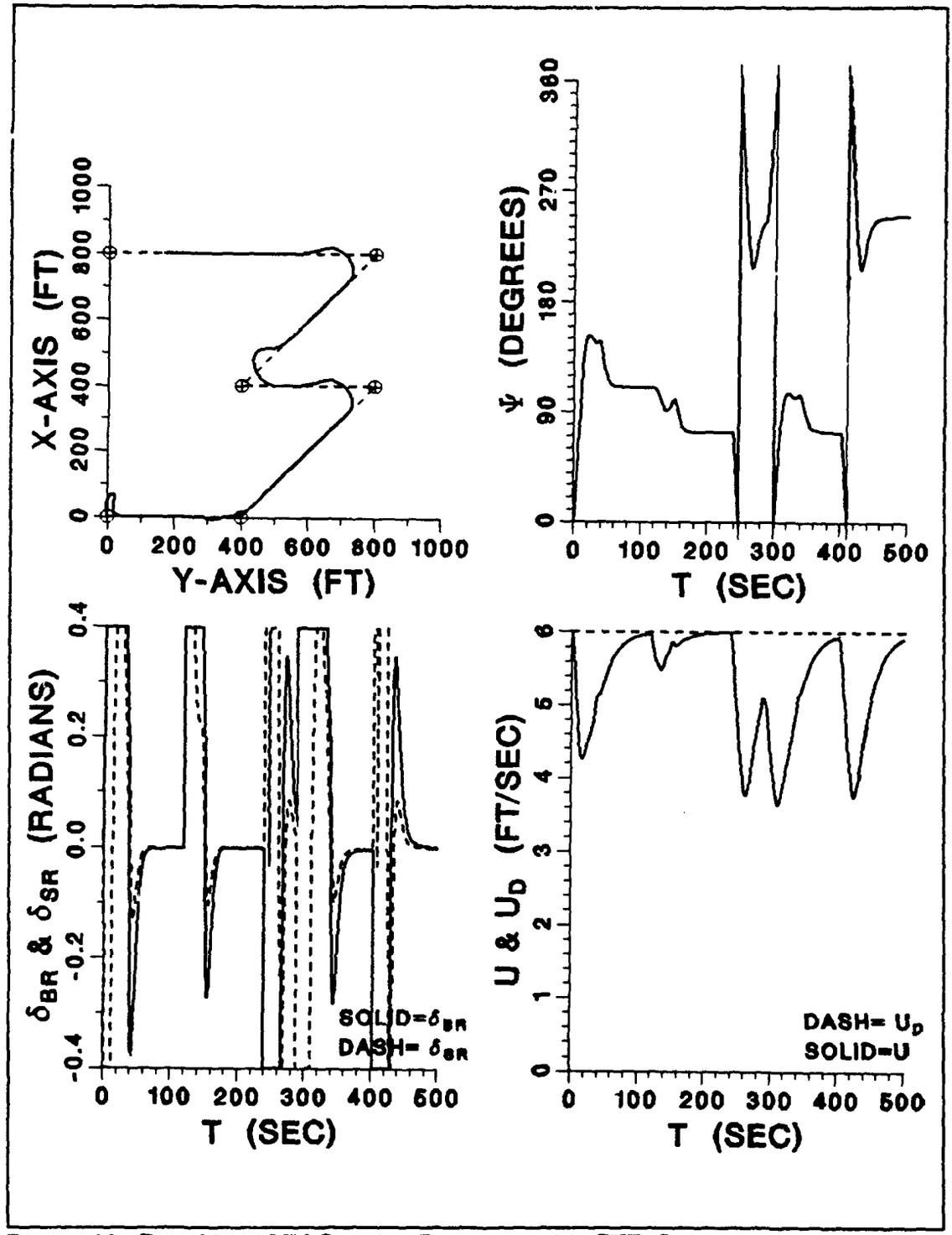


Figure 45. Run 22 - AUV Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, Hit = 7L)

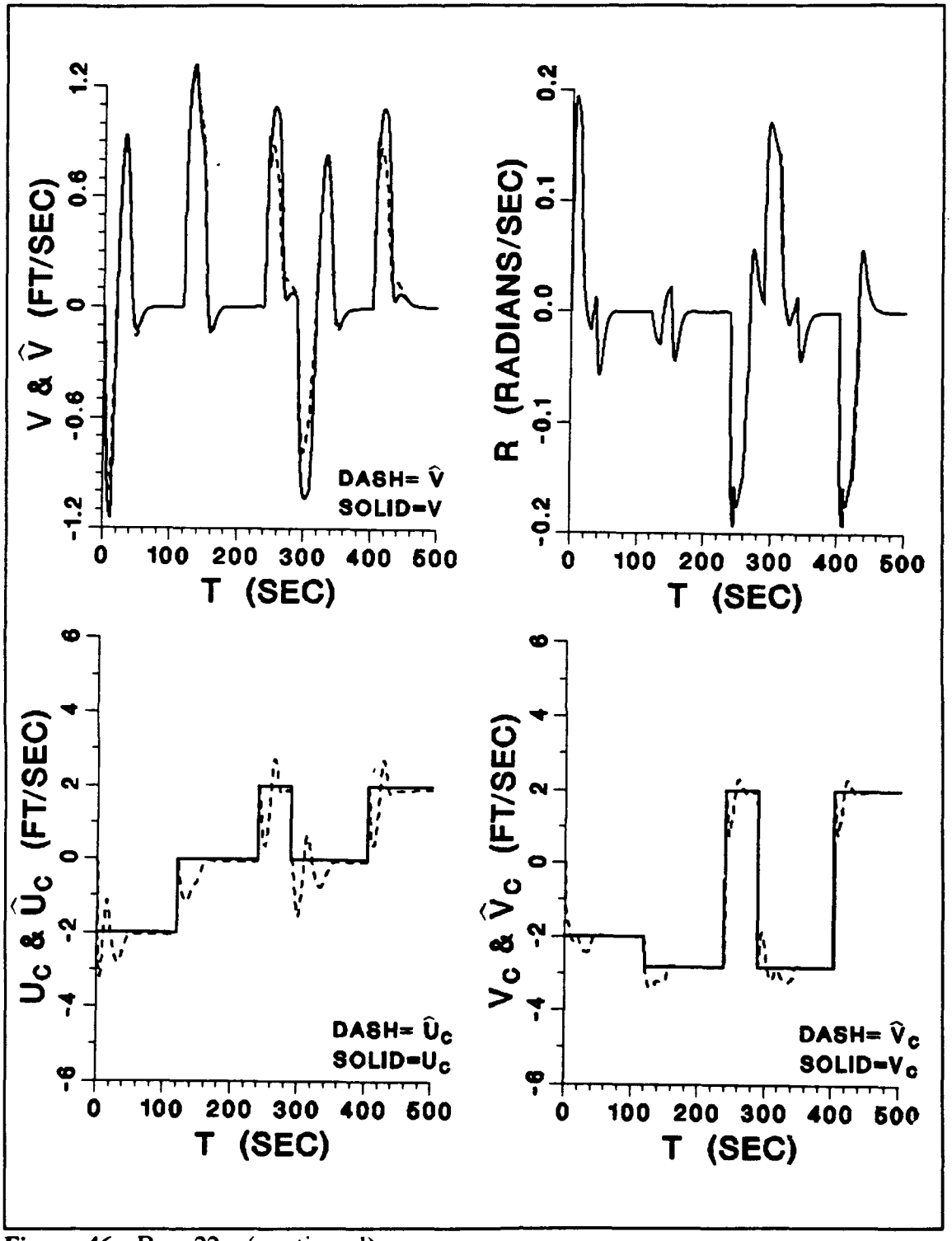


Figure 46. Run 22 - (continued)

compromise. There are alternate methods for modifying the AUV's control package, but these are beyond the scope of this thesis and are the subject of further study at NPS.

3. Theory of Rudder Action Compensation

As alluded in Chapter IV, an alternate method for steady-state disturbance compensation requires the AUV's rudders to overcome the underwater current, as opposed to using a heading error in the previous method. This capability is an inherent advantage of a MIMO control system. A SISO steering controller could not achieve such action.

The design starts with new steady-state assumptions:

$$r = \psi' = y' = 0 \tag{5.16}$$

The CTE equation for $\dot{y}'$ simplifies to:

$$v = -v_c \tag{5.17}$$

and using (5.17) in (5.1) yields:

$$\begin{aligned} b_{11}\delta_{br} + b_{12}\delta_{sr} &= a_{11}v_c \\ b_{21}\delta_{br} + b_{22}\delta_{sr} &= a_{21}v_c \end{aligned} \tag{5.18}$$

Treating (5.18) as two equations with two unknowns, we solve for the steady-state rudder deflections needed to overcome the underwater current:

$$\delta_{br} = \left(\frac{a_{11}b_{22} - a_{21}b_{12}}{b_{11}b_{22} - b_{21}b_{12}} \right) v_c$$
$$\delta_{sr} = \left(\frac{a_{21}b_{11} - a_{11}b_{21}}{b_{11}b_{22} - b_{21}b_{12}} \right) v_c \tag{5.19}$$

As before, we must analyze the resultant effect on the sliding surfaces and control laws at steady state. Substitution of (5.19) into the control laws (5.5) allows the determination of the uncompensated sliding surfaces at steady-state in the presence of an underwater current:

$$\sigma_1 = \frac{v_c \phi}{\eta^2} \left[\frac{(k_{11}k_{26} - k_{16}k_{21})}{(k_{15}k_{26} - k_{16}k_{25})} + \frac{k_{26}(a_{11}b_{22} - a_{21}b_{12}) - k_{16}(a_{21}b_{11} - a_{11}b_{21})}{(k_{15}k_{26} - k_{16}k_{25})(b_{11}b_{22} - b_{12}b_{21})} \right]$$
$$\sigma_2 = \frac{v_c \phi}{\eta^2} \left[\frac{(k_{11}k_{25} - k_{15}k_{21})}{(k_{16}k_{25} - k_{15}k_{26})} + \frac{k_{25}(a_{11}b_{22} - a_{21}b_{12}) - k_{15}(a_{21}b_{11} - a_{11}b_{21})}{(k_{16}k_{25} - k_{15}k_{26})(b_{11}b_{22} - b_{12}b_{21})} \right] \tag{5.20}$$

Writing (5.20) as:

$$\sigma_1 = \sigma_1^*$$
$$\sigma_2 = \sigma_2^* \tag{5.21}$$

Substituting (5.21) into the sliding surface equations (5.4) results in two equations for the uncompensated CTE at steady-state:

$$y' = \frac{\sigma_1^* + s_{11} v_c}{s_{14}}$$
$$y' = \frac{\sigma_2^* + s_{21} v_c}{s_{24}} \tag{5.22}$$

Finally, we compensate the sliding surface equations to counteract (5.21) and (5.22):

$$\sigma_1 = s_{11} v + s_{12} r + s_{13} \psi' + s_{14} y' + \sigma_1^* + s_{11} v_c$$
$$\sigma_2 = s_{21} v + s_{22} r + s_{23} \psi' + s_{24} y' + \sigma_2^* + s_{21} v_c \tag{5.23}$$

4. Simulation with Rudder Action Compensation

To analyze the effectiveness of rudder disturbance compensation, Run 23 was conducted at u = 6.0 ft/s, ϕ = 2.0, and η^2 = 0.35 with a local lateral current of v_c = -1.0 ft/s. Figures 47 and 48 show Run 23 which was produced by simulation program SDVCTELOSRR300OBS_LQR.FOR (see Appendix A) and DISSPLA plotting program PLOT10.FOR. To easily demonstrate the effectiveness of the compensation, the simulation starts at ψ = 45° at way-point (0,0) headed towards the next way-point at (1000,1000). At t = 0, the lateral current is initiated on the AUV. The vehicle parameters oscillate slightly as the observers and controller compensate for the step in current. It's the steady-state values that must be closely scrutinized. The AUV settles out at approximately ψ = 45° as desired! The rudders are both deflected in the same direction to combat the current. The bow rudder is saturated at δ_{br} = 0.4 and the stern rudder is at δ_{sr} = 0.3. If the bow rudder did not saturate,

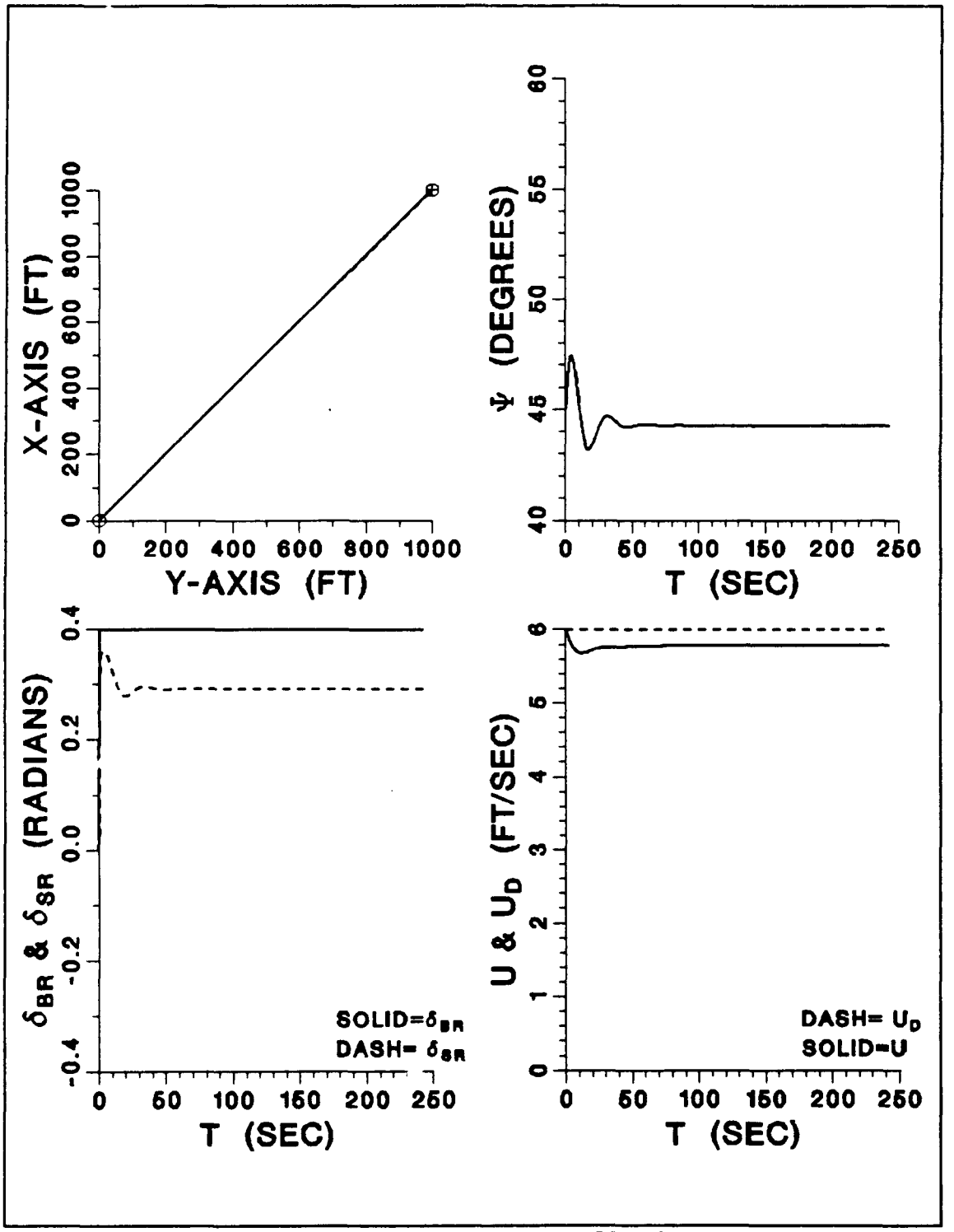


Figure 47. Run 23 - AUV Steering Response with CTE Guidance, (Disturbance Compensation by Rudder Action, v_c = -1.0 ft/s)

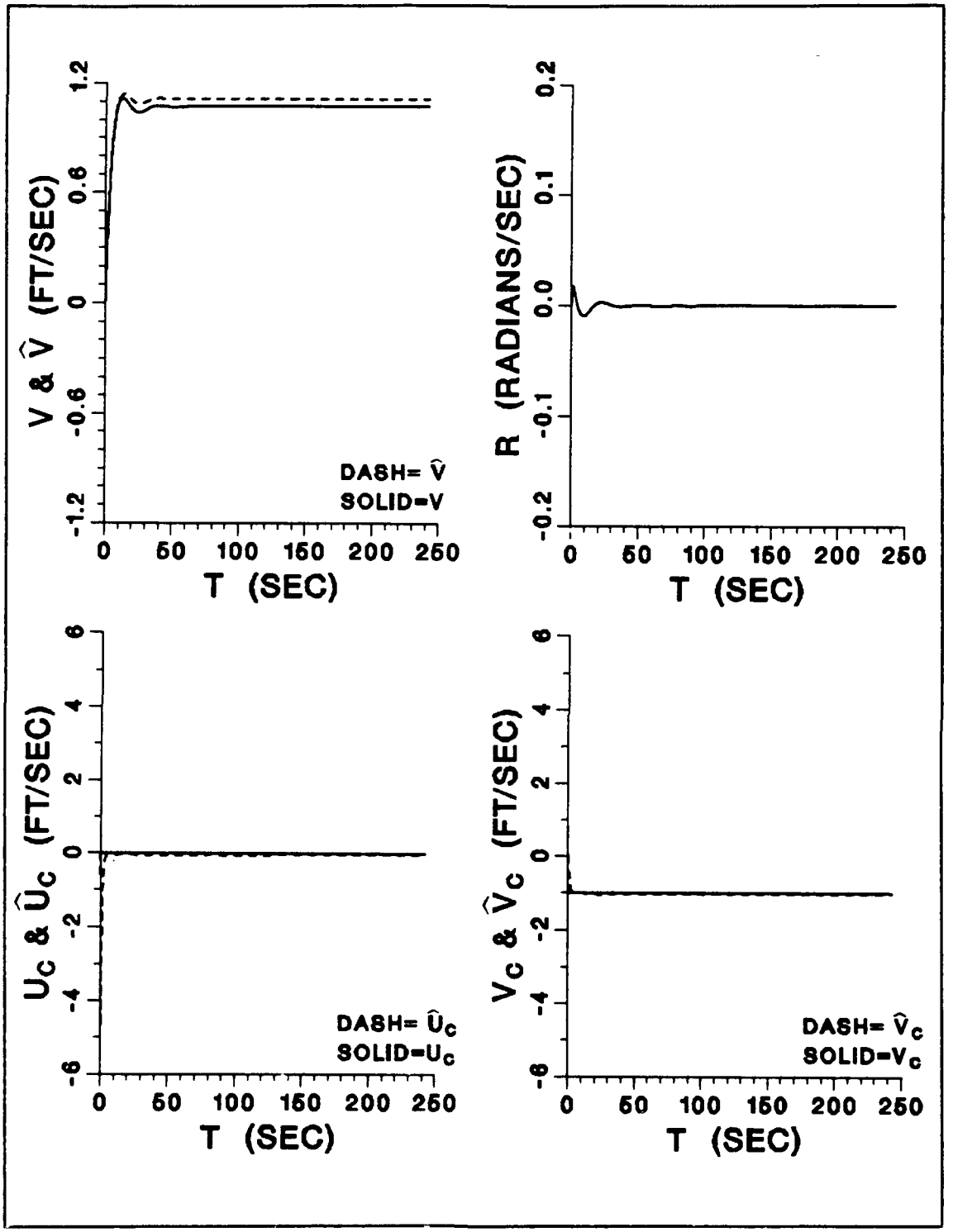


Figure 48. Run 23 - (continued)

the AUV would achieve a heading angle of exactly 45°. The difference in the rudder deflections is both due to the forward-aft asymmetry of the AUV's (actually the SDV's) body and the coefficients in the control laws. Note that the sway velocity v settles out at about 1.0 ft/s as predicted by (5.17).

For comparison, Run 24 on Figures 49 and 50 have been included. Run 24 was conducted with the same conditions as Run 23, but the controller in Run 24 uses heading error disturbance compensation discussed previously. Note that the rudders have zero deflection at steady-state, but the AUV settles out at about $\psi = 55°$ which is a 10° heading error. Hence in comparing Runs 23 and 24, rudder disturbance compensation eliminates a 10° heading error.

Taking a further look at Run 23 we note again that with a lateral current of only -1.0 ft/s, one rudder is saturated and the other is close to saturation. Obviously, we expect the controller will be unable to compensate for a larger lateral current. Run 25 was conducted to investigate the response of the CTE controller with the rudder method of disturbance rejection in the presence of a larger current of v_c = -2.0 ft/s. It's interesting to note on Figures 51 and 52 of Run 25 that the controller does the best job it can of reducing the heading error. The rudders are deflected over, but to achieve zero cross-track-error the AUV must also have a heading error of approximately 10° ($\psi = 55°$).

Run 26, shown on Figures 53 and 54, was included for comparison with Run 25. The controller used in Run 26 uses heading error disturbance compensation. Note that the AUV settles out at a heading of about $\psi = 65°$ for a heading error of

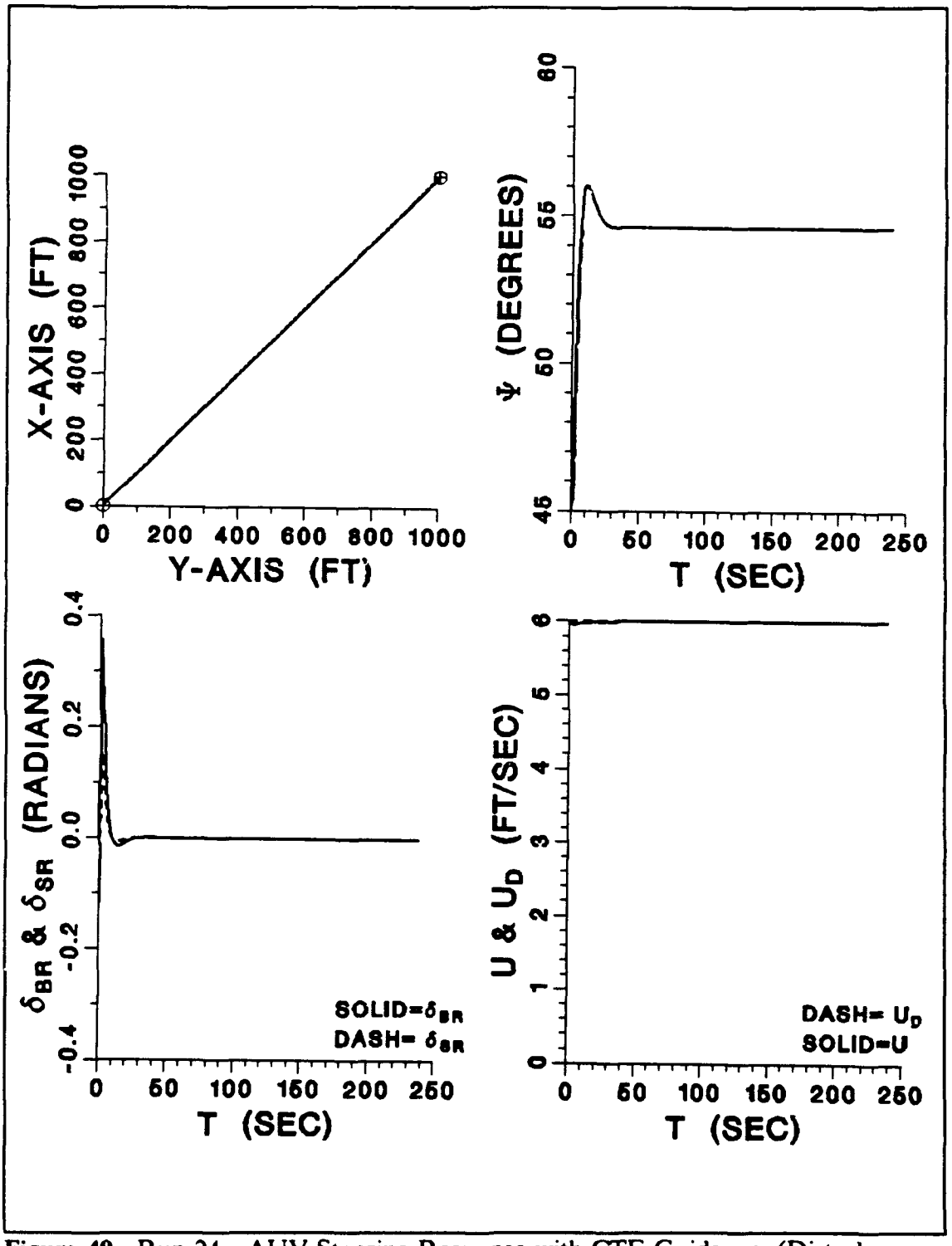


Figure 49. Run 24 - AUV Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, v_c = -1.0 ft/s)

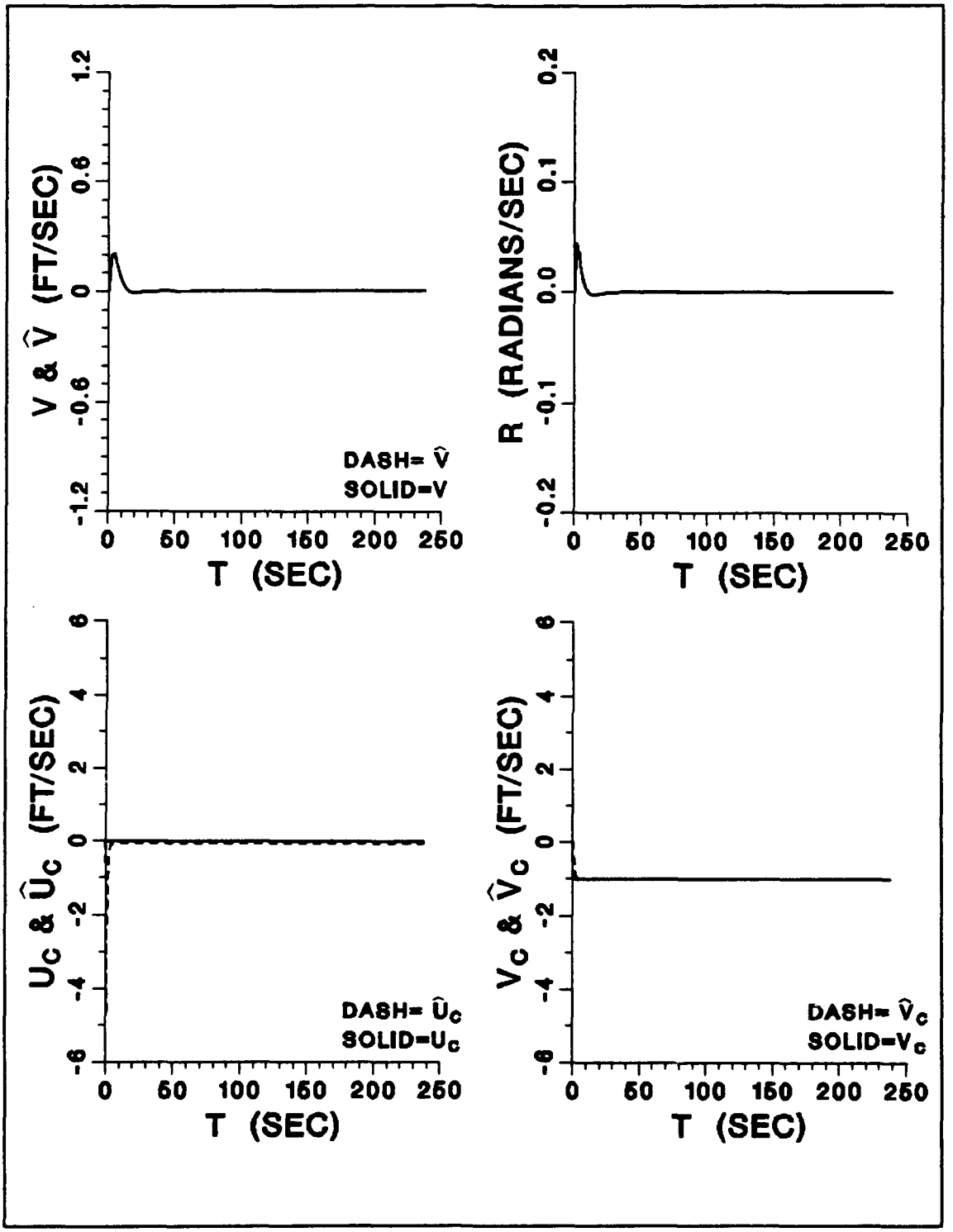


Figure 50. Run 24 - (continued)

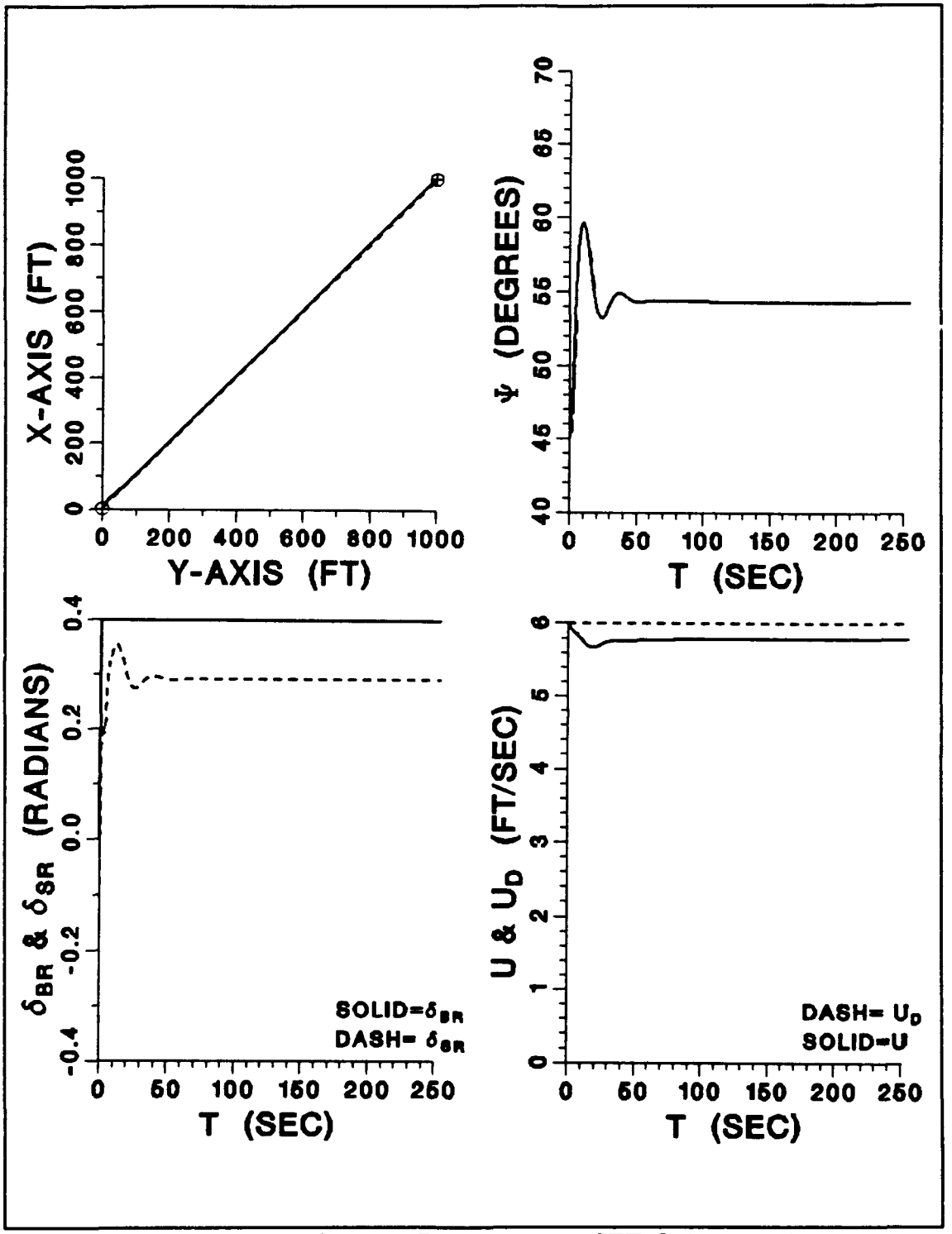


Figure 51. Run 25 - AUV Steering Response with CTE Guidance, (Disturbance Compensation by Rudder Action, v_c = -2.0 ft/s)

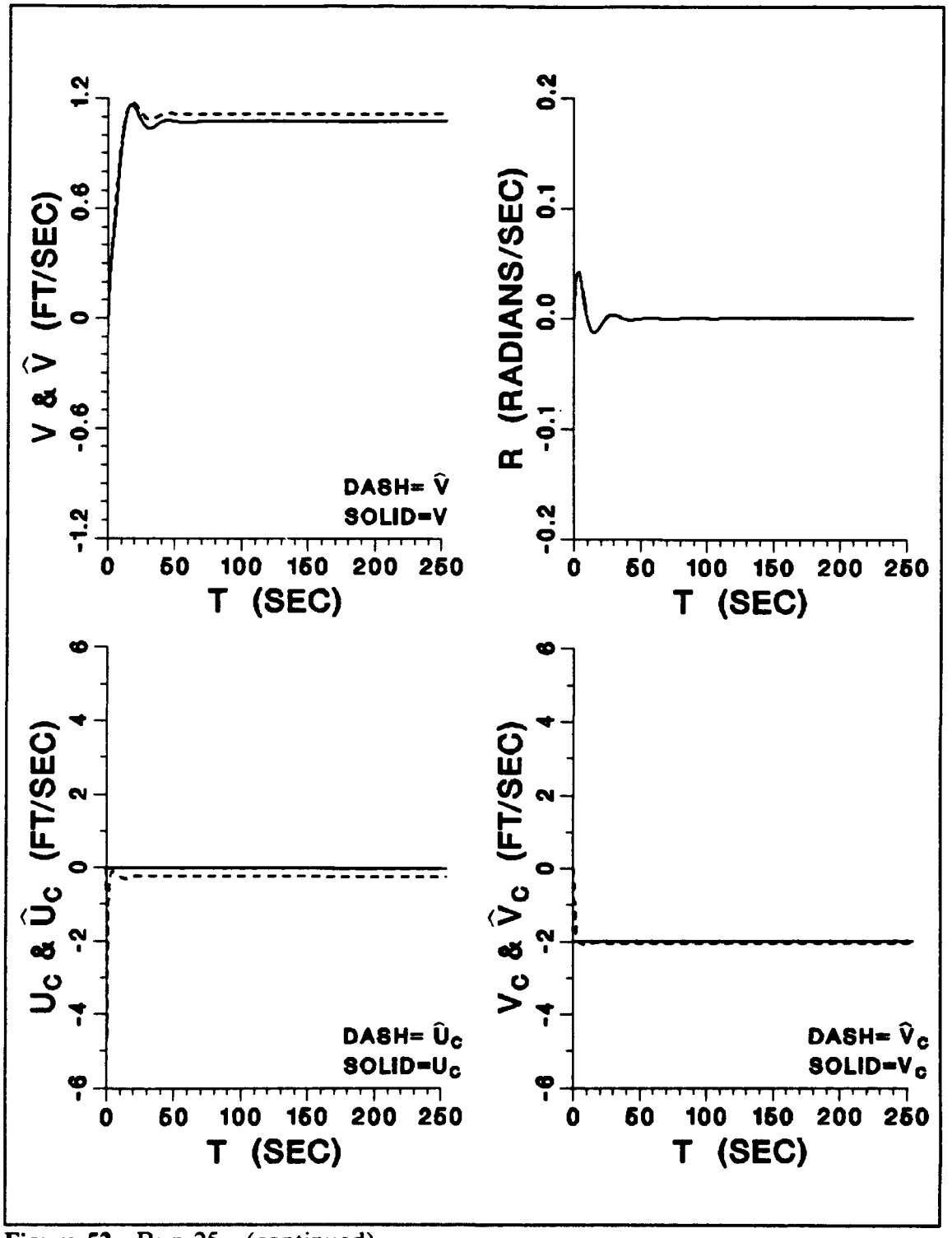


Figure 52. Run 25 - (continued)

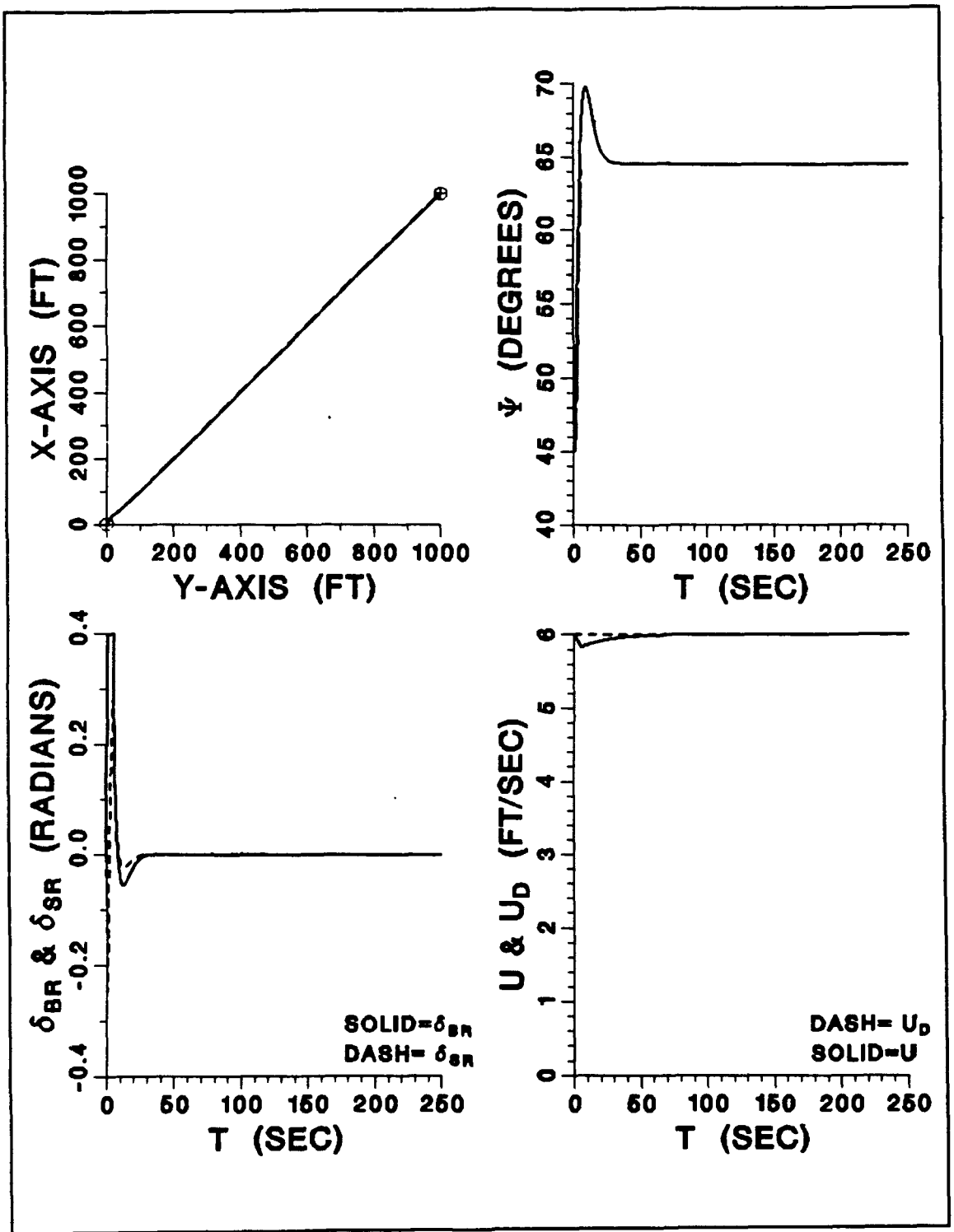


Figure 53. Run 26 - AUV Steering Response with CTE Guidance, (Disturbance Compensation by Heading Error, v_c = -2.0 ft/s)

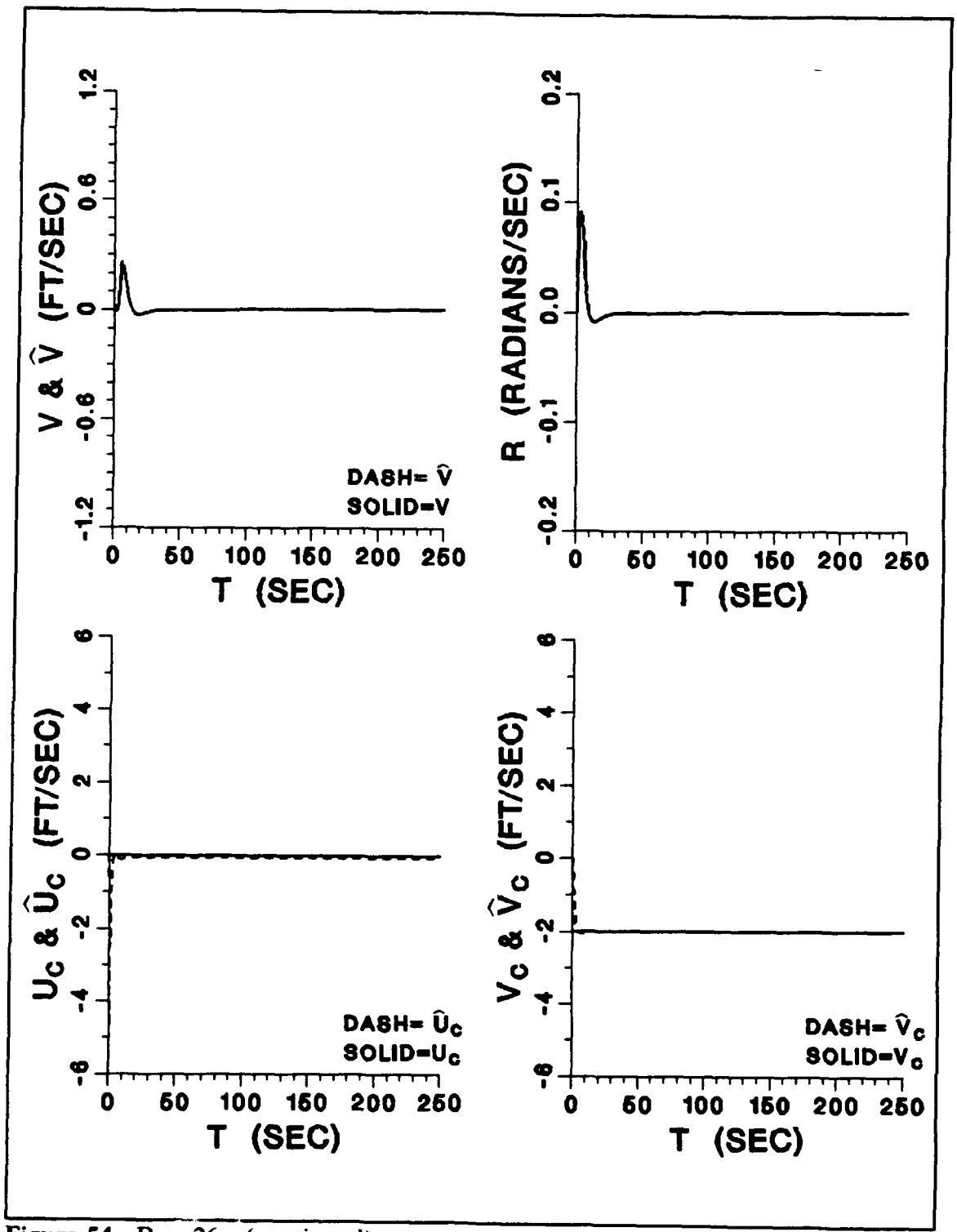


Figure 54. Run 26 - (continued)

about 20°. Thus, the CTE controller with rudder disturbance compensation, although unable to fully eliminate the heading error, does reduce it significantly.

E. CONCLUDING REMARKS

A cross-track-error guidance steering controller has been designed for the AUV which also uses the observers designed in Chapter IV. The CTE controller effectively follows a path in the horizontal plane. One difficulty is the criterion for a hit of a way-point. If the distance for a hit is too small when the next leg must be reached through a large angle, significant overshoot occurs after passing through the way-point. If too large a distance is used when the following leg must be reached through a shallow angle, the AUV heads for the next leg too early. Thus, the hit distance is a compromise reached through analysis of the AUV's ordered speed and prospective track.

It should be emphasized that in the MIMO design adopted here, the sway velocity v assumes significant values and, therefore, becomes important in the control law design. This is in contrast to SISO designs where v is negligibly small and can be neglected.

Disturbance compensation was designed into the controller by adding a term to each sliding surface. Two methods of disturbance compensation were utilized. First, the AUV overcame the current with a heading error. The advantage of heading error compensation is its effectiveness over a wide range of underwater

currents. The second method of rudder compensation will keep the AUV pointed along the track, but it's unable to compensate for lateral currents above 1.0 ft/s.

The final step of this thesis is to apply the steering and diving controllers, which were designed separately, to the full nonlinear model.

VI. AUV SIMULATION IN THREE DIMENSIONAL SPACE

A. INTRODUCTION

In previous chapters, simulation of the AUV was conducted with only simplified nonlinear equations of motion. To prove the validity of the diving and steering controllers which were designed based on these simplified nonlinear equations of motion, control of the AUV must be simulated using the full nonlinear equations developed by Smith, Crane, and Summey [Ref. 7]. First, the AUV will be controlled with the diving and LOS controllers. Second the CTE controller will be used vice the LOS controller. Finally, three dimensional path keeping will be developed and simulated.

B. SIMULTANEOUS LOS STEERING AND DEPTH CONTROL

Figures 55 and 56 show Run 27 which was produced by program SDV3D_LOS.FOR and DISSPLA program PLOT3D_LOS.FOR (see Appendix A). The diving controller was executed with ϕ = 2.0 and η^2 = 0.2, the LOS controller with ϕ = 2.0 and η^2 = 0.3, and the speed controller with ϕ = 1.0. Ordered speed was u_D = 6.0 ft/s and global currents were u_{co} = -2.0 ft/s and v_{co} = 2.0 ft/s. The hit criterion was one vehicle length. Perfect state feedback was used.

Figures 55 and 56 show that the controllers do an excellent job of vehicle control even during simultaneous depth and course changes. Movement of the

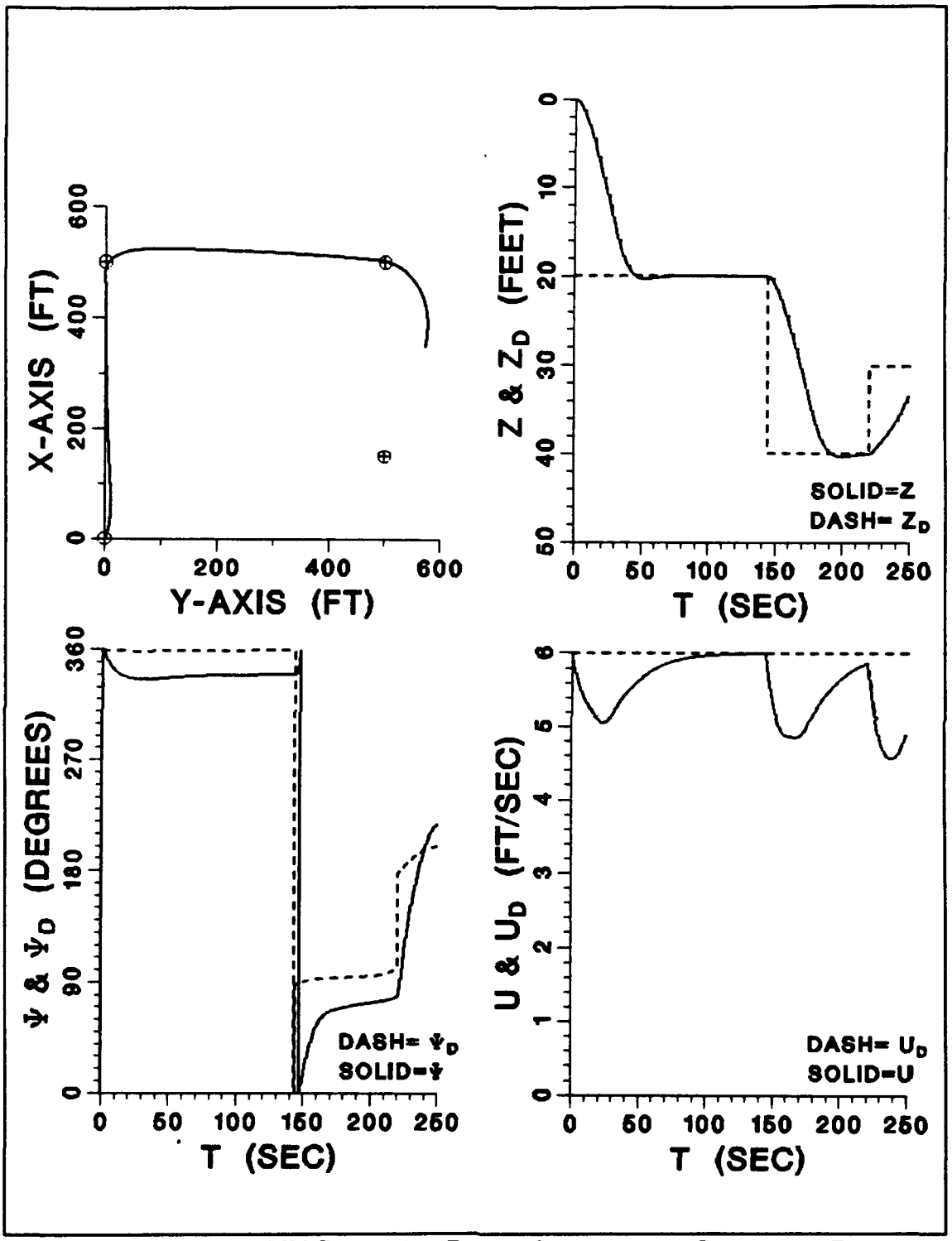


Figure 55. Run 27 - AUV Simulation During Simultaneous Course and Depth Changes, LOS Steering Controller, Perfect State Feedback

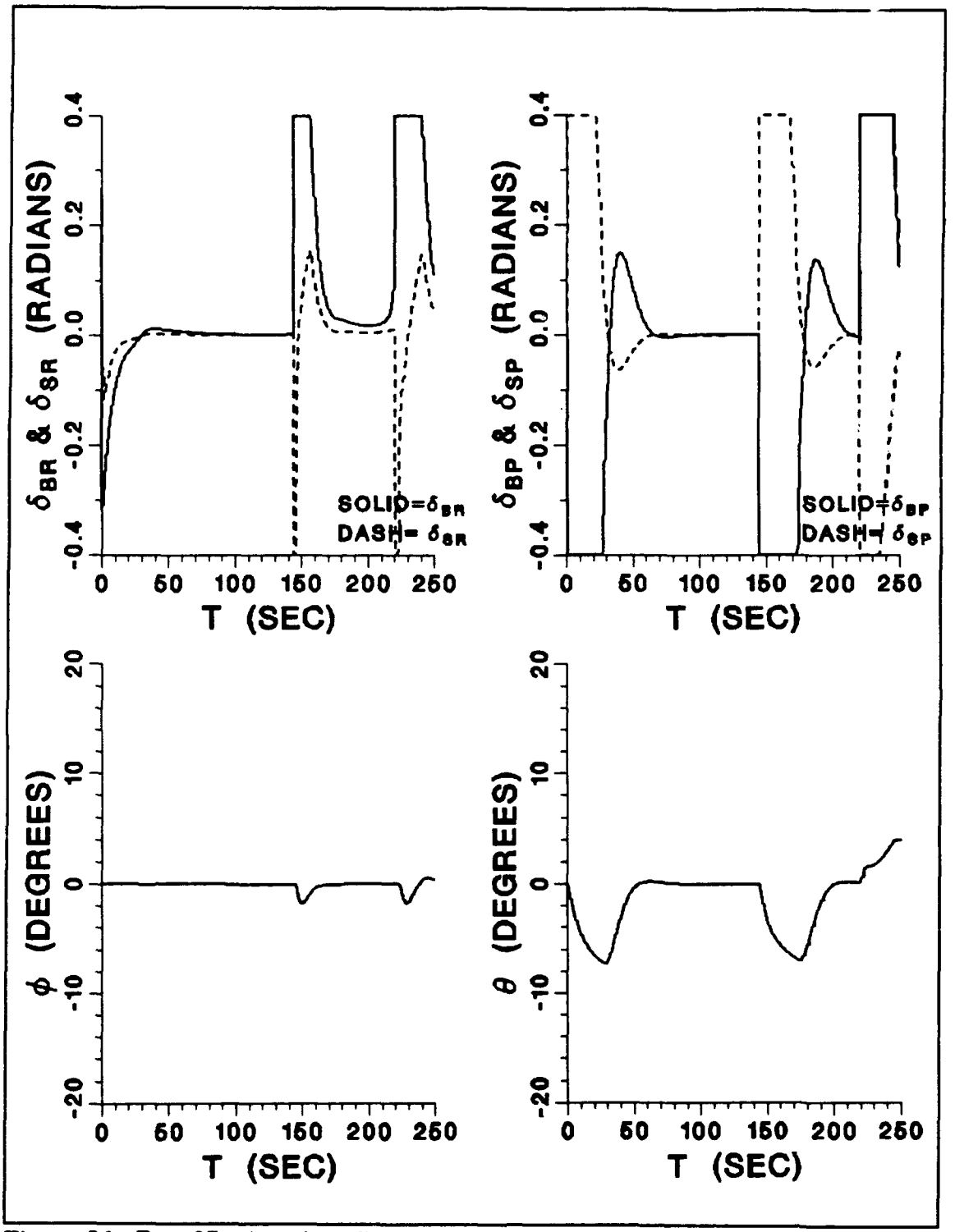


Figure 56. Run 27 - (continued)

control surfaces is smooth with no oscillation. Maximum vehicle roll is less than 2°. Of course, the AUV does overshoot way-point (500,500) due to the combined effect of the small hit criterion and the v_{co} component of the current. The reader should note that the simulation ends before the vehicle reaches the final ordered way-point.

The conclusion is that a combined control package containing the diving and LOS steering controllers is effective for AUV control during simultaneous course and depth changes.

C. SIMULTANEOUS CTE STEERING AND DEPTH CONTROL

Figures 57 and 58 show Run 28 which was produced by program SDV3D_CTE.FOR and DISSPLA program PLOT3D_CTE.FOR (see Appendix A). The only change from the previous run was the CTE steering controller was executed with $\eta^2 = 0.35$ and the hit criterion was increased to six vehicle lengths.

Figures 57 and 58 reveal that the CTE steering and diving controllers are effective in controlling the AUV. The movement of the rudders is not as smooth as the previous run with the LOS controller. This is primarily due to the CTE guidance scheme which requires strong control action to achieve minimal y'. Note that the increased control surface action results in a maximum roll angle of about 5° which is 3° larger than Run 27.

D. THREE DIMENSIONAL VEHICLE PATH KEEPING

So far, path keeping in the horizontal plane has been coupled with depth keeping in the vertical plane. More complex obstacle avoidance and path planning

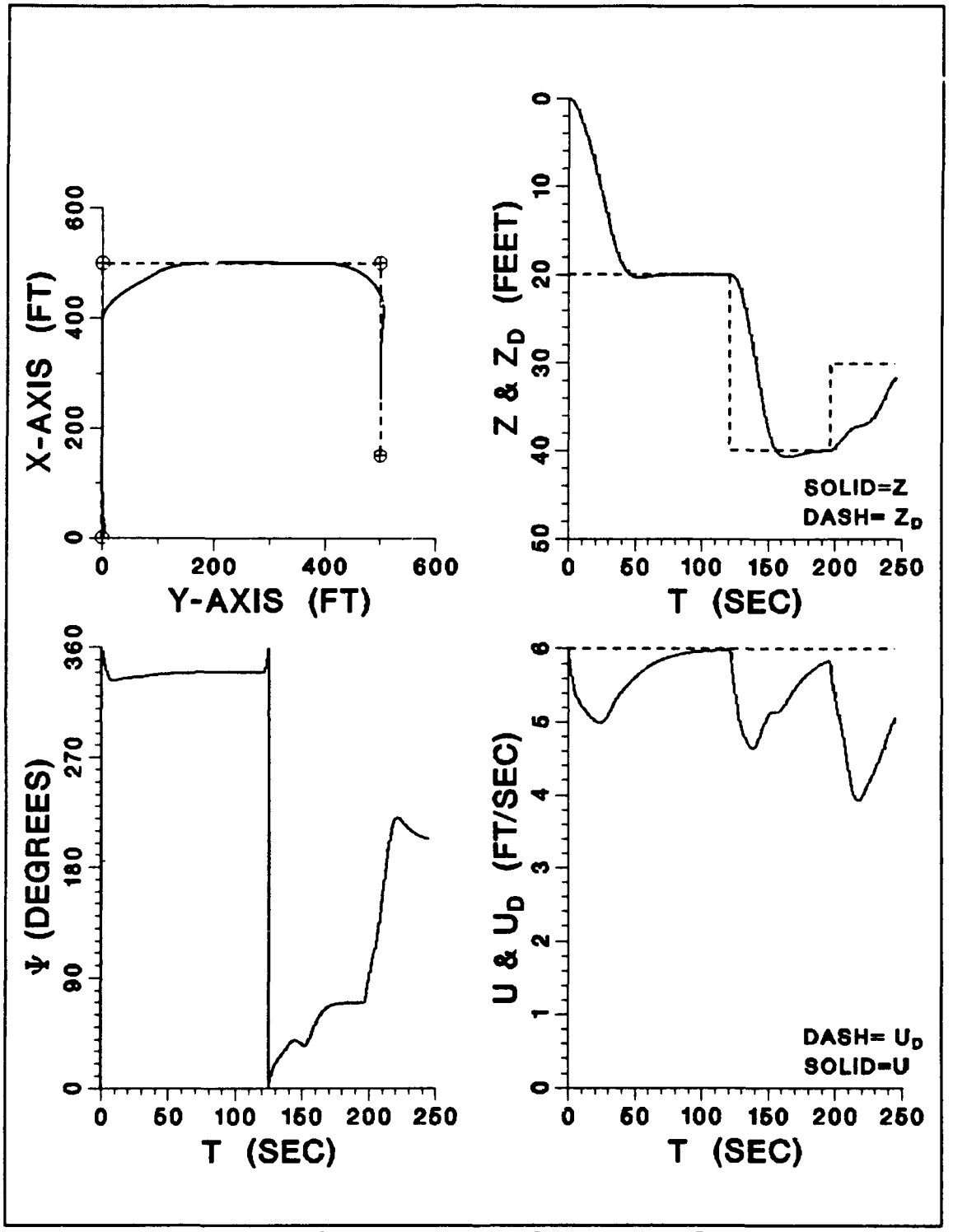


Figure 57. Run 28 - AUV Simulation During Simultaneous Course and Depth Changes, CTE Steering Controller, Perfect State Feedback

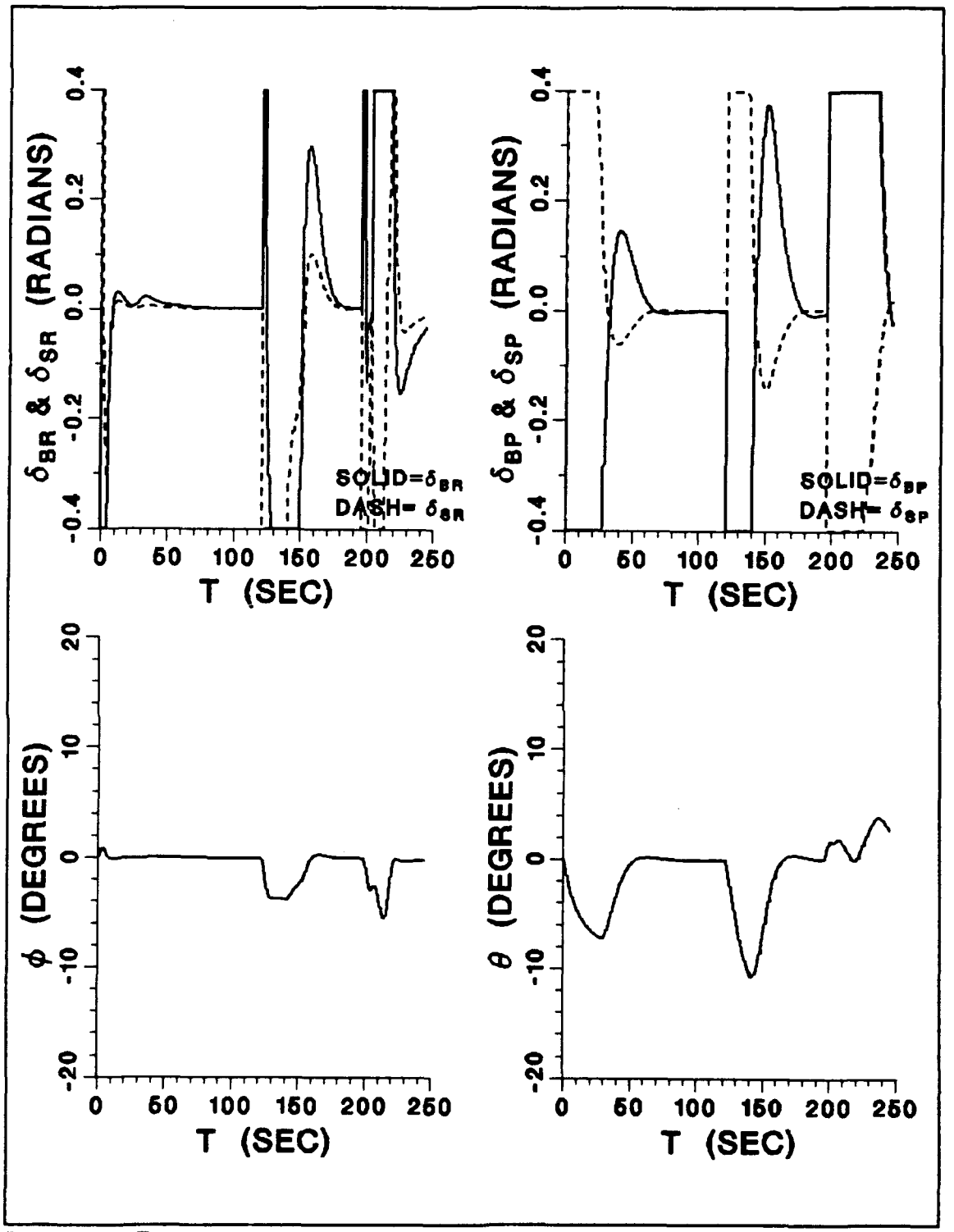


Figure 58. Run 28 - (continued)

situations may, however, require that the vehicle can accurately follow a path defined in a three dimensional coordinate frame. To this end, we must first consider the problem of path keeping in the vertical plane.

1. Path Keeping in the Vertical Plane

The linearized equations of motion in the vertical plane of a neutrally buoyant symmetric vehicle with $x_G = x_B = 0$, $z_B = 0$, and $z_G > 0$ are:

$$(m - Z_{\dot{w}})\dot{w} - Z_{\dot{q}}\dot{q} = (Z_q + m)uq + Z_w uw + Z_{\delta_{sp}} u^2 \delta_{sp} + Z_{\delta_{bp}} u^2 \delta_{bp} \quad (6.1)$$

$$(I_y - M_{\dot{q}})\dot{q} - M_{\dot{w}}\dot{w} = M_q uq + M_w uw - z_G mg\sin\theta + M_{\delta_{sp}} u^2 \delta_{sp} + M_{\delta_{bp}} u^2 \delta_{bp} \quad (6.2)$$

$$\dot{\theta} = q \quad (6.3)$$

$$\dot{x} = u\cos\theta + w\sin\theta \quad (6.4)$$

$$\dot{z} = -u\sin\theta + w\cos\theta \quad (6.5)$$

where the pitch angle θ is not necessarily small, and the coordinate system is shown in Figure 59. The assumption that $z_G > 0$ ensures that the vehicle is stable in roll, which is normally the case. If β denotes the commanded pitch angle, positive nose up, we rotate the coordinate system as shown in Figure 59 and we get:

$$z' = z\cos\beta + x\sin\beta \quad (6.6)$$

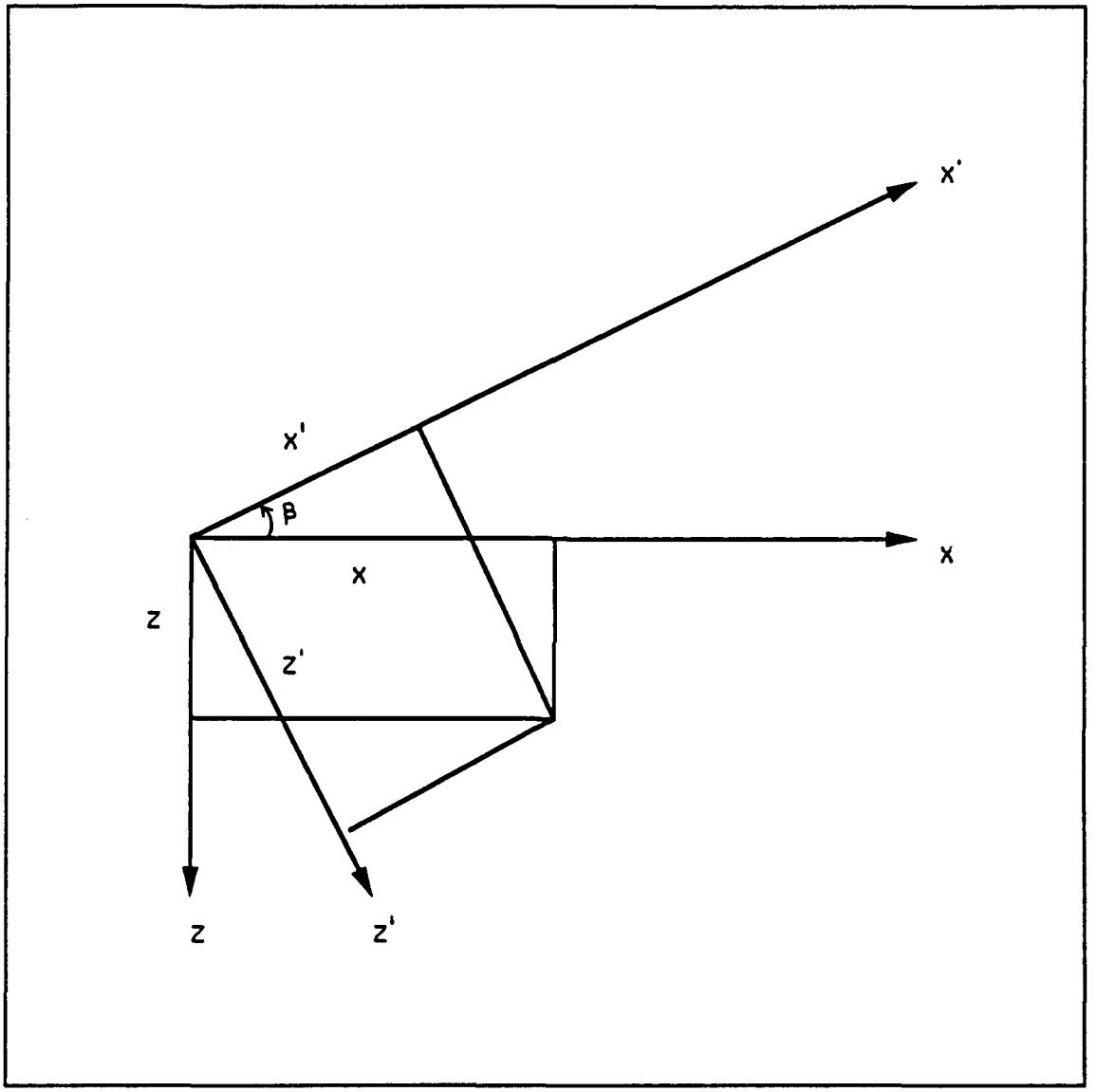


Figure 59. Coordinate System in the Vertical Plane

$$x' = -z \sin\beta + x \cos\beta \tag{6.7}$$

$$\theta = \beta + \theta' \tag{6.8}$$

where θ' is a small angle, defined as the deviation between the actual and the commanded pitch angle, and z' is the cross-track-error off the assumed straight line path. Then the linearized equations of motion in the vertical plane in the (x', z') system of coordinates become:

$$\dot{\theta}' = q \tag{6.9}$$

$$\dot{w} = a_{11} uw + a_{12} uq + b_1 u^2 \delta \tag{6.10}$$

$$\dot{q} = a_{21} uw + a_{22} uq + a_{23} \theta' + b_2 u^2 \delta + d \tag{6.11}$$

$$\dot{z}' = -u\,\theta' + w \tag{6.12}$$

where we have set $\delta_{sp} = \delta$, $\delta_{bp} = 0$ for now, and the coefficients are given by:

$$a_{11} = \frac{(I_y - M_{\dot{q}}) Z_w + Z_q M_w}{(m - Z_{\dot{w}})(I_y - M_{\dot{q}}) - Z_q M_{\dot{w}}} \tag{6.13}$$

$$a_{12} = \frac{(I_y - M_{\dot{q}})(Z_q + m) + Z_q M_q}{(m - Z_{\dot{w}})(I_y - M_{\dot{q}}) - Z_q M_{\dot{w}}} \tag{6.14}$$

$$a_{21} = \frac{M_w Z_w + (m - Z_w) M_w}{(m - Z_w)(I_y - M_q) - Z_q M_w} \tag{6.15}$$

$$a_{22} = \frac{M_w (Z_q + m) + (m - Z_w) M_q}{(m - Z_w)(I_y - M_q) - Z_q M_w} \tag{6.16}$$

$$a_{23} = \frac{-z_G mg(m - Z_w)\cos\beta}{(m - Z_w)(I_y - M_q) - Z_q M_w} \tag{6.17}$$

$$b_1 = \frac{(I_y - M_q) Z_\delta + Z_q M_\delta}{(m - Z_w)(I_y - M_q) - Z_q M_w} \tag{6.18}$$

$$b_2 = \frac{M_w Z_\delta + (m - Z_w) M_\delta}{(m - Z_w)(I_y - M_q) - Z_q M_w} \tag{6.19}$$

$$d = \frac{-z_G mg(m - Z_w)\sin\beta}{(m - Z_w)(I_y - M_q) - Z_q M_w} \tag{6.20}$$

The sliding mode control law for system (6.9) through (6.12) is expressed as:

$$\delta = k_1 \theta' + k_2 w + k_3 q + k_n satsgn(\sigma) \tag{6.21}$$

$$\sigma = s_1 \theta' + s_2 w + s_3 q + s_4 z' + s_5 \tag{6.22}$$

The gains k_1, k_2, k_3 and the sliding plane coefficients s_1, s_2, s_3, s_4 can be computed using either the LQR or pole placement techniques as developed in Chapter II. It should be mentioned that since the coefficient a_{23} in (6.17) depends on the commanded pitch angle β, the above gains depend on β as well. Therefore, they

have to be recomputed every time a new angle β is commanded. It was found, however, that their variation in β is small and a constant average angle β can be used in the control law design.

The extra coefficient s_5 in (6.22) appears because of the nonzero metacentric height, or the constant term d in (6.11). This feedforward term s_5 can be computed from steady-state accuracy requirements. At steady-state, equations (6.9) through (6.12) yield:

$$\delta = \frac{a_{11}d}{a_{21}b_1u^2 + a_{23}b_1 - a_{11}b_2u^2} \tag{6.23}$$

$$\theta' = \frac{-b_1d}{a_{21}b_1u^2 + a_{23}b_1 - a_{11}b_2u^2} \tag{6.24}$$

$$w = \frac{-b_1ud}{a_{21}b_1u^2 + a_{23}b_1 - a_{11}b_2u^2} \tag{6.25}$$

Equation (6.23) provides the dive plane angle that is necessary to achieve the commanded rise or dive angle, and equations (6.24) and (6.25) reveal that this is only possible with a nonzero heave velocity and path orientation error. Then, equation (6.21) yields:

$$satsgn(\sigma) = \frac{(a_{11} + k_1b_1 + k_2b_1u)d}{k_n(a_{21}b_1u^2 + a_{23}b_1 - a_{11}b_2u^2)} \tag{6.26}$$

and using:

$$satsgn(\sigma) = \frac{\sigma}{\phi} \tag{6.27}$$

together with the requirement that the cross-track-error z' reaches zero at steady-state, we get from (6.22):

$$s_5 = \frac{(a_{11} + k_1 b_1 + k_2 b_1 u)\phi d + k_n(s_1 + s_2 u) b_1 d}{k_n(a_{21} b_1 u^2 + a_{23} b_1 - a_{11} b_2 u^2)} \tag{6.28}$$

Equation (6.27) is valid if $|satsgn(\sigma)| \leq 1$, and this yields the critical value of k_n for stability:

$$k_n \geq \left| \frac{(a_{11} + k_1 b_1 + k_2 b_1 u) d}{a_{21} b_1 u^2 + a_{23} b_1 - a_{11} b_2 u^2} \right| \tag{6.29}$$

The above scheme will provide the desired response once the vehicle hydrodynamics and geometric properties are accurately known. In cases of actual vehicle/mathematical model mismatch, a steady-state path error will develop. This is especially true for different vehicle loading conditions which change the value of the metacentric height from its nominal design value. An integrator in z' (with a sufficiently large time constant to avoid undesirable oscillations) or an estimator of z_G can be incorporated into the design to ensure the required steady-state accuracy. For typical cases of the SDV applications considered in this work, this did not seem to be necessary since the actual steady-state path errors appear to be very small.

This is demonstrated in Figure 60. The vehicle is initially horizontal and the commanded way-point is located at $(x, y) = (15, -15)$ ship lengths, which corresponds to a 45° rise angle. Curve 1 simulates the nominal design with $z_G = 0.1$ ft, while Curve 2 simulates the case where the actual z_G is three times larger than the assumed value, and in the case of Curve 3 it is five times smaller. It can be seen that even under such a wide variation, the actual path error is negligible.

When both bow and stern planes are acting, the linearized equations (6.9) through (6.12) become:

$$\dot{\theta}' = q \tag{6.30}$$

$$\dot{w} = a_{11}uw + a_{12}uq + b_{11}u^2\delta_{sp} + b_{12}u^2\delta_{bp} \tag{6.31}$$

$$\dot{q} = a_{21}uw + a_{22}uq + b_{21}u^2\delta_{sp} + b_{22}u^2\delta_{bp} + a_{23}\theta' + d \tag{6.32}$$

$$\dot{z}' = -u\theta' + w \tag{6.33}$$

and the control laws:

$$\delta_{sp} = k_{11}\theta' + k_{12}w + k_{13}q + k_{14}\eta^2 satsgn(\sigma_1) + k_{15}\eta^2 satsgn(\sigma_2) \tag{6.34}$$

$$\delta_{bp} = k_{21}\theta' + k_{22}w + k_{23}q + k_{24}\eta^2 satsgn(\sigma_1) + k_{25}\eta^2 satsgn(\sigma_2) \tag{6.35}$$

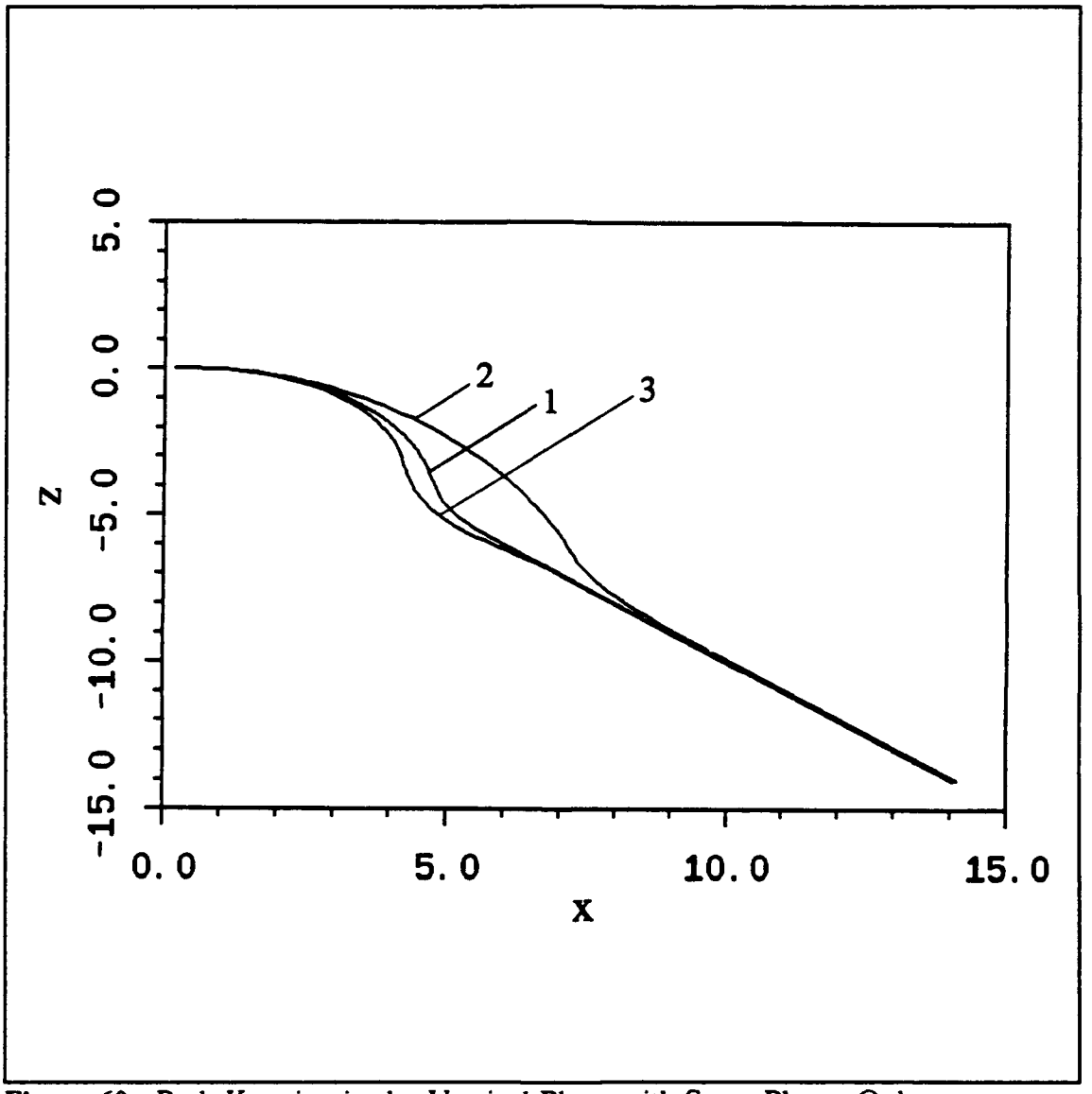


Figure 60. Path Keeping in the Vertical Plane with Stern Planes Only (z_G = 0.1 ft (nominal), z_G = 0.3 ft, and z_G = 0.5 ft)

with:

$$\sigma_1 = s_{11}\theta' + s_{12}w + s_{13}q + s_{14}z' + s_{15} \tag{6.36}$$

$$\sigma_2 = s_{21}\theta' + s_{22}w + s_{23}q + s_{24}z' + s_{25} \tag{6.37}$$

The existence of two independent inputs can now ensure zero steady-state path deviation with zero heave velocity and the commanded pitch angle as well. Equations (6.30) through (6.33) yield at steady-state:

$$q = \theta' = w = 0 \tag{6.38}$$

$$\delta_{sp} = \frac{b_{12}d}{(b_{11}b_{22} - b_{12}b_{21})u^2} \tag{6.39}$$

$$\delta_{bp} = \frac{b_{11}d}{(b_{11}b_{22} - b_{12}b_{21})u^2} \tag{6.40}$$

Then, using equations (6.34) through (6.37) we get:

$$s_{15} = \frac{(b_{12}k_{25} + b_{11}k_{15})\phi d}{\eta^2 u^2 (b_{11}b_{22} - b_{12}b_{21})(k_{14}k_{25} - k_{15}k_{24})} \tag{6.41}$$

$$s_{25} = \frac{-(b_{11}k_{14} + b_{12}k_{24})\phi d}{\eta^2 u^2 (b_{11}b_{22} - b_{12}b_{21})(k_{14}k_{25} - k_{15}k_{24})} \tag{6.42}$$

and the control law is complete. Results for the SDV at u = 2.5 ft/s, using stern planes only (Curves 1) and combined stern and bow planes (Curves 2) are presented in Figures 61 and 62. It can be clearly seen that the application of the multiple input sliding mode technique leads to a much better steady-state accuracy.

2. Three Dimensional Path Keeping

Cross-track-error controls in the horizontal and vertical planes can be coupled together to provide accurate vehicle path keeping in three dimensions. The related geometry is illustrated in Figure 63 along with some nomenclature. The angles α_H and α_V are the two coordinate rotation angles to align a local frame to the desired vehicle path. At the same time, α_V is the vehicle commanded pitch angle. The horizontal plane cross-track-error y' is controlled by the rudders while the vertical plane cross-track-error z' is driving the dive planes. The two coordinate rotations are shown in Figure 64 where the horizontal plane rotation is executed first, and then the vertical plane rotation follows. The geometric relations that describe the above rotations are:

$$\alpha_H = \arctan\left(\frac{y_D - y_o}{x_D - x_o}\right) \tag{6.43}$$

$$x' = (y - y_o)\sin\alpha_H + (x - x_o)\cos\alpha_H \tag{6.44}$$

$$y' = (y - y_o)\cos\alpha_H - (x - x_o)\sin\alpha_H \tag{6.45}$$

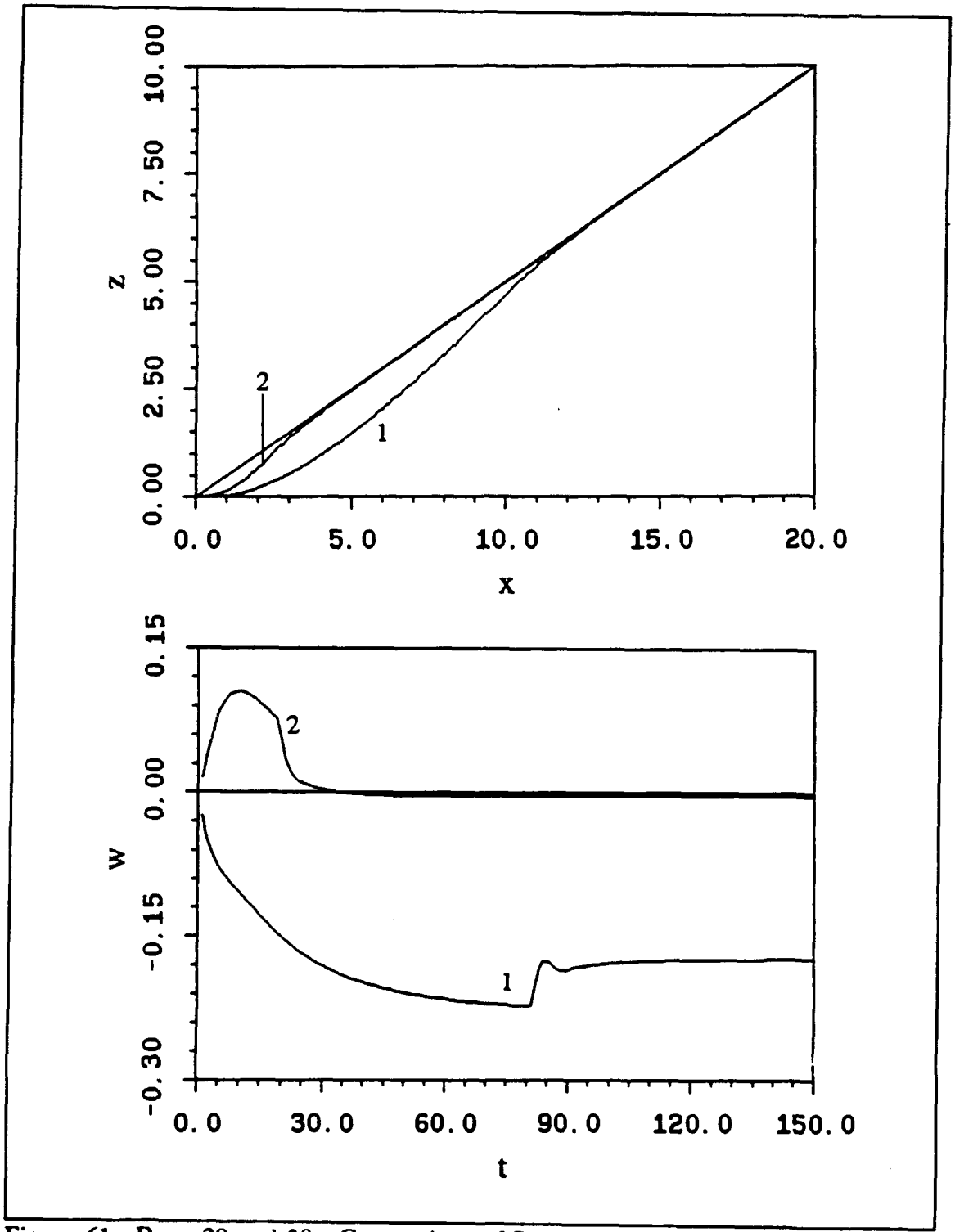


Figure 61. Runs 29 and 30 - Comparison of Path Keeping in the Vertical Plane with Stern Planes Only (Curves 1) and Stern & Bow Planes (Curves 2)

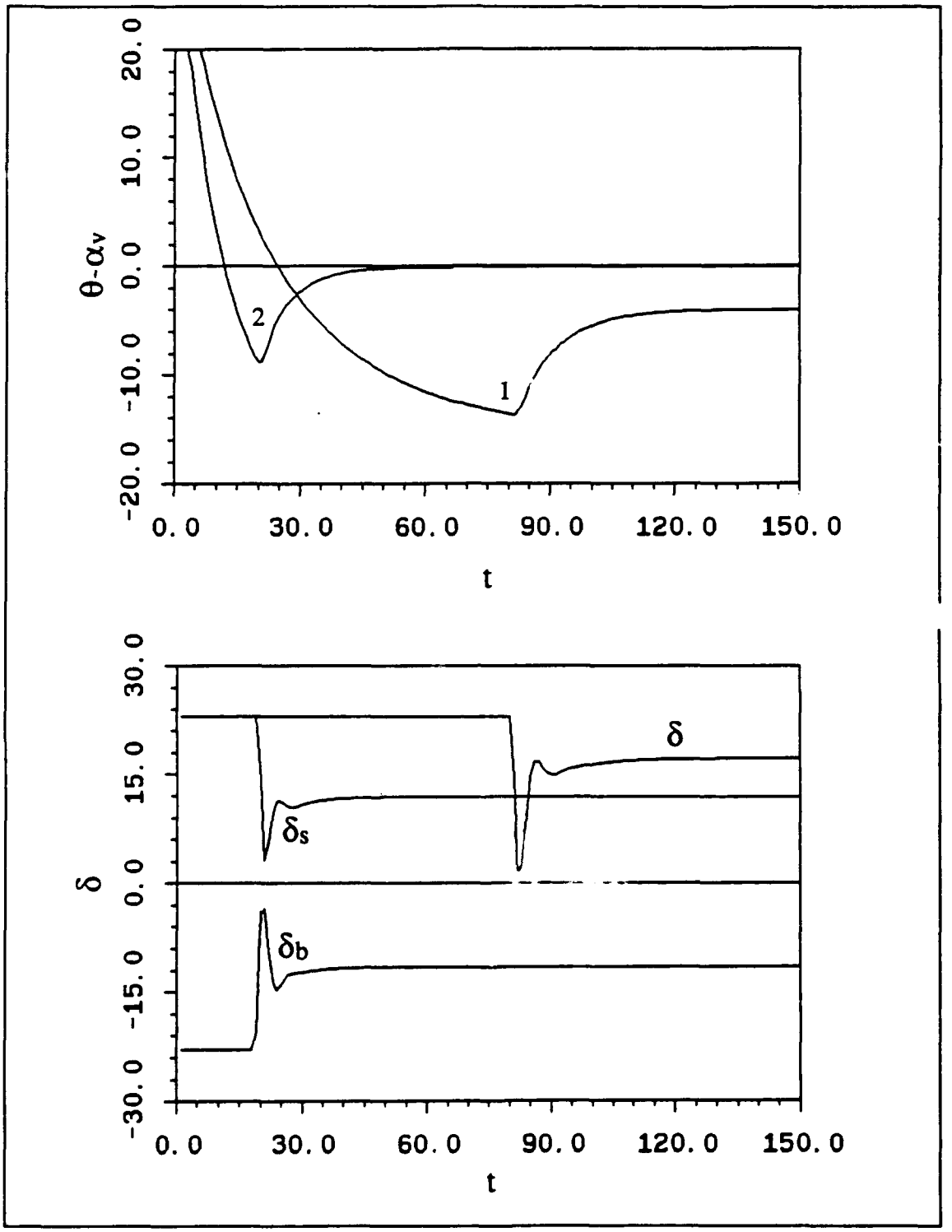
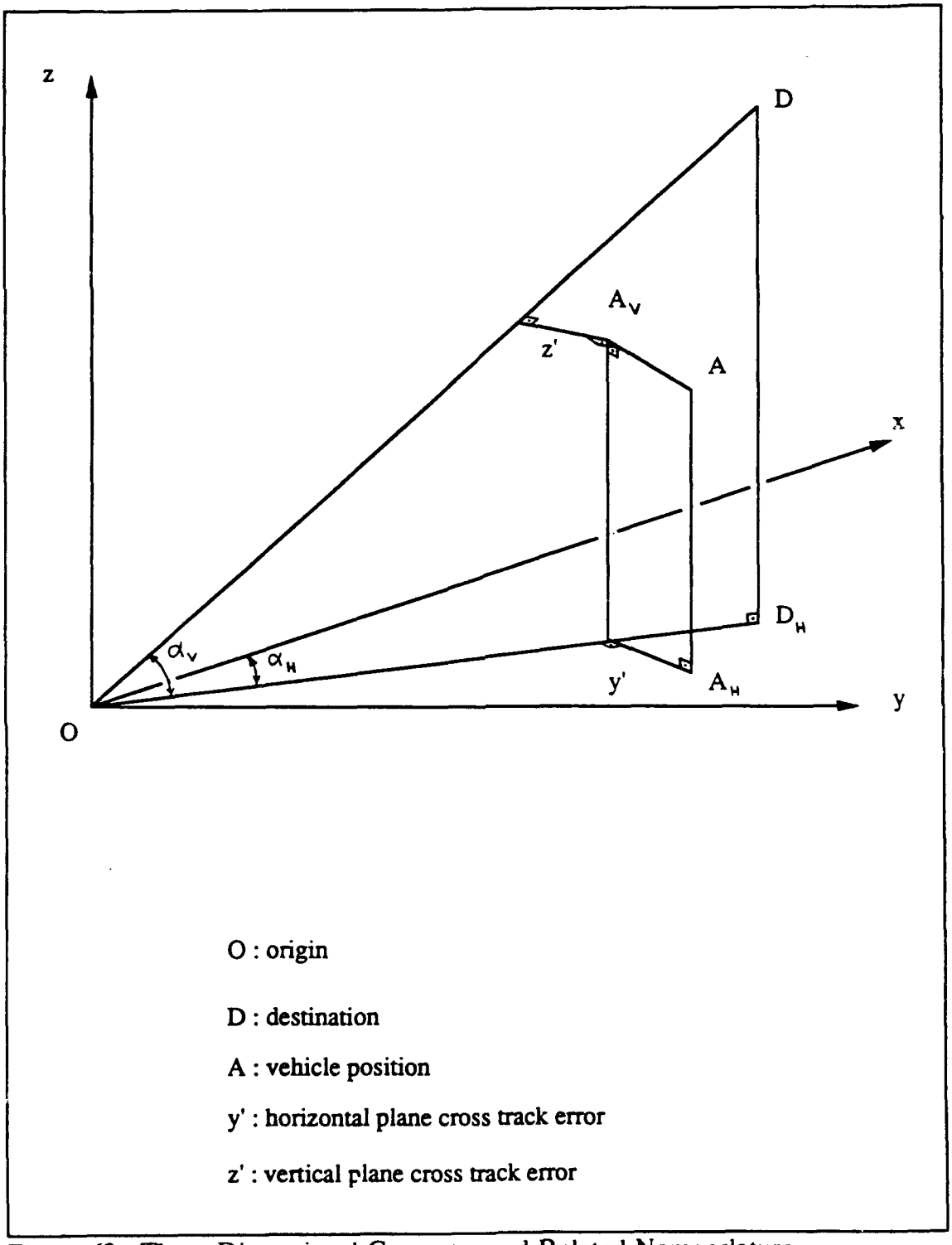


Figure 62. Runs 29 and 30 - (continued)



O : origin

D : destination

A : vehicle position

y' : horizontal plane cross track error

z' : vertical plane cross track error

Figure 63. Three Dimensional Geometry and Related Nomenclature

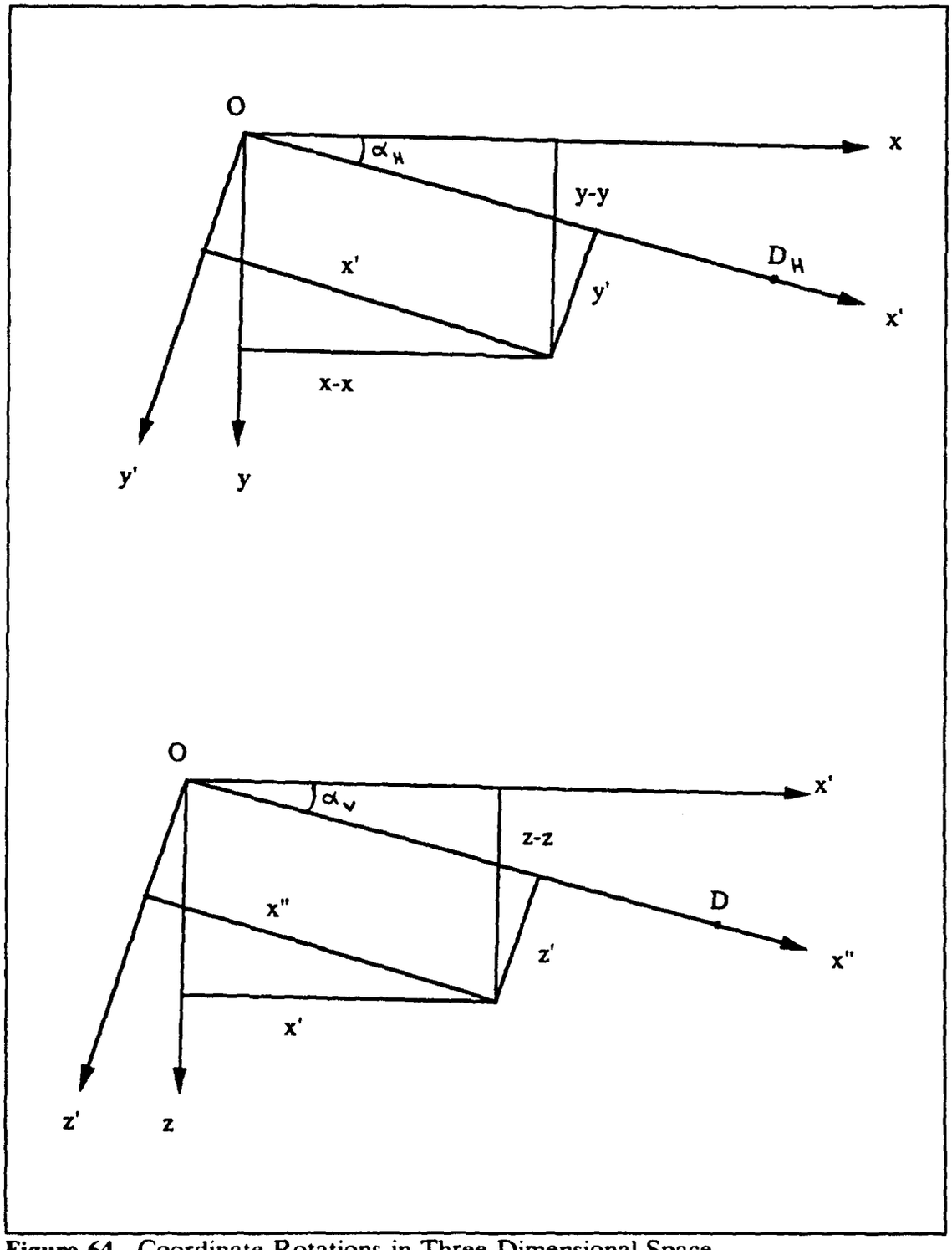


Figure 64. Coordinate Rotations in Three Dimensional Space

$$\alpha_V = \arctan\left(\frac{z_D - z_o}{x'_D}\right) \tag{6.46}$$

$$x'_D = (y_D - y_o)\sin\alpha_H + (x_D - x_o)\cos\alpha_H \tag{6.47}$$

$$x'' = -(z - z_o)\sin\alpha_V + x'\cos\alpha_V \tag{6.48}$$

$$z' = (z - z_o)\cos\alpha_V + x'\sin\alpha_V \tag{6.49}$$

Results are presented for the following way-points $(x, y, z) = (10, 0, 5)$, $(20, 5, 5)$, $(30, -5, -3)$, $(50, 0, -5)$ vehicle lengths, using one stern rudder and independent stern and bow planes as developed before, in Figures 65 through 67. The commanded vehicle forward speeds are 3.0, 2.0, and i.5 ft/s, respectively, and the target distance was fixed at 2 ship lengths. The commanded and actual vehicle paths are plotted projected on the three planes (x, y), (x, z), and (y, z). As can be seen the above coordinate rotations provide excellent path keeping capabilities except for the low speed of 1.5 ft/s where the dive planes cannot always provide the necessary dive and rise moment. At such low speeds, however, the vertical thrusters are to be turned on, thus providing additional heave force and/or pitch moment.

E. CONCLUDING REMARKS

The diving and steering controllers designed in previous chapters were combined to perform simultaneous depth and steering control. Both the LOS and

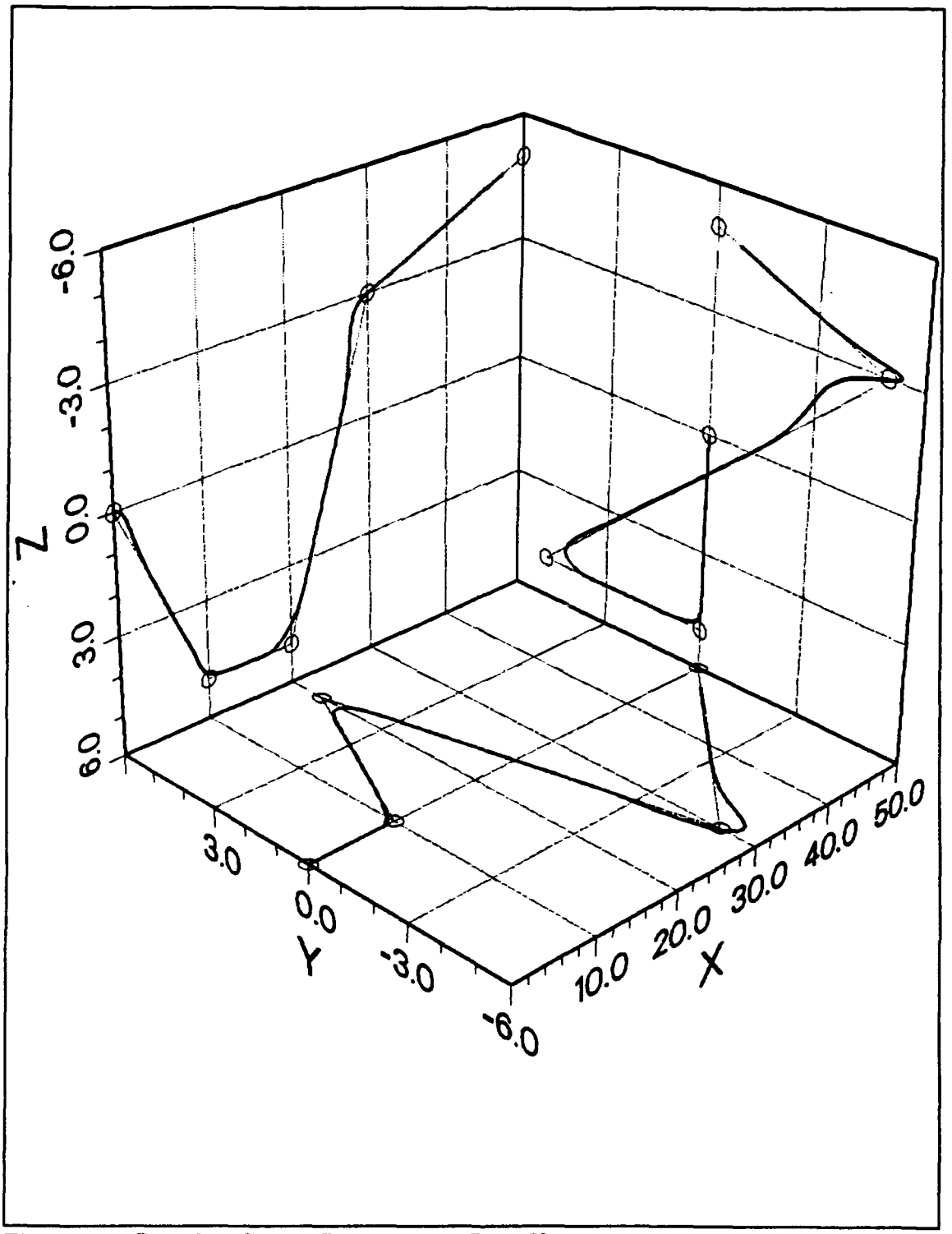


Figure 65. Run 31 - Three Dimensional Path Keeping at u = 3.0 ft/s

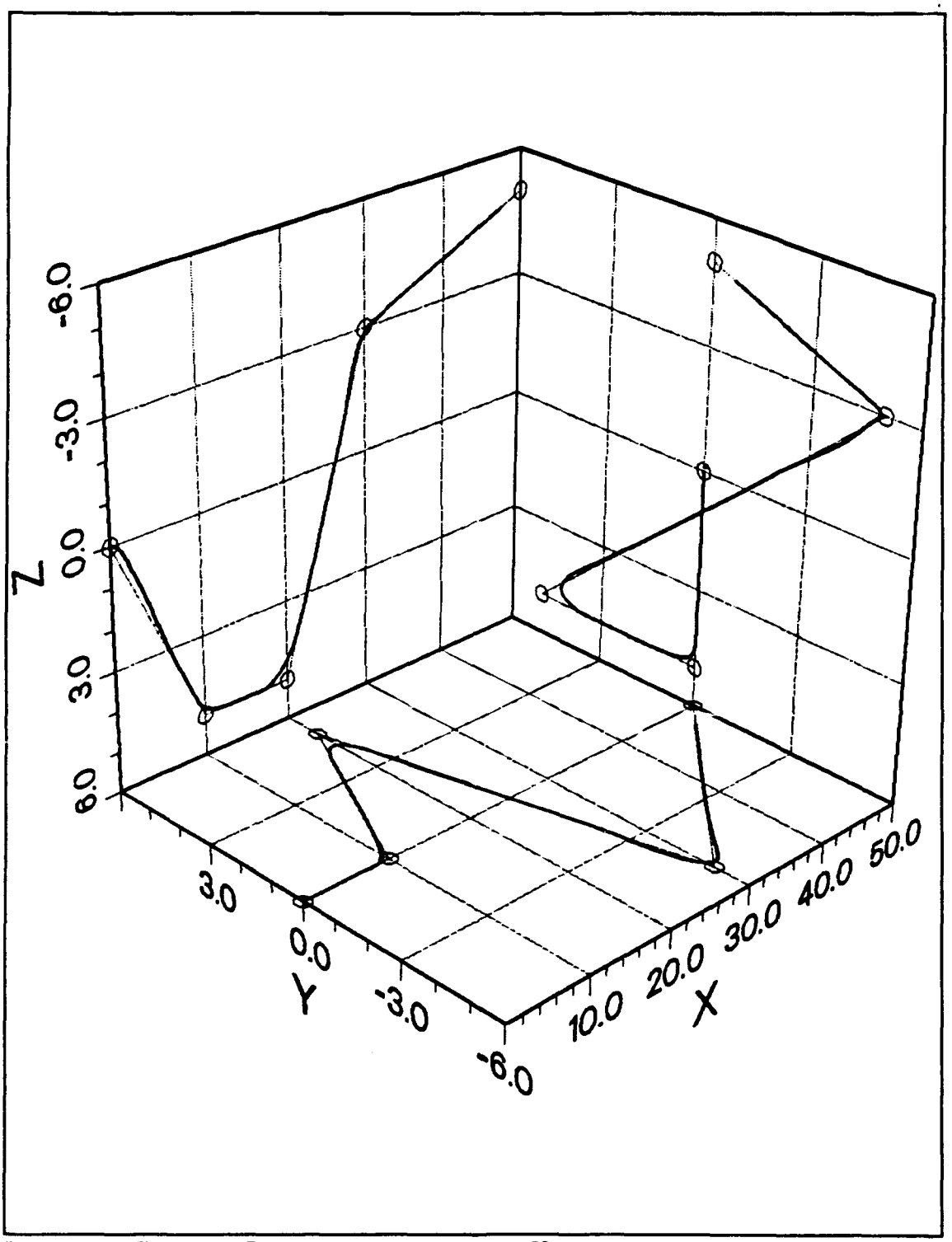


Figure 66. Run 32 - Three Dimensional Path Keeping at u = 2.0 ft/s

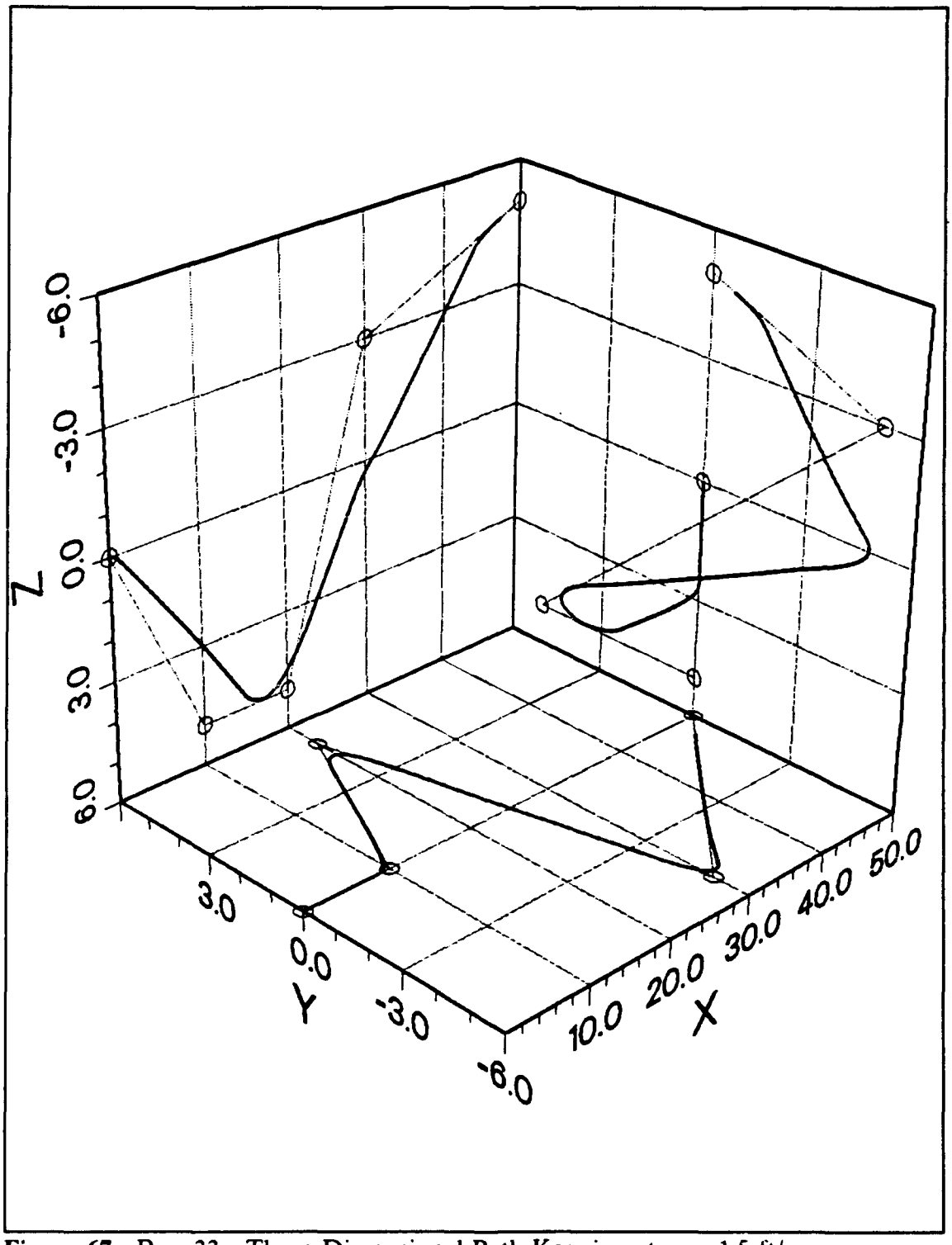


Figure 67. Run 33 - Three Dimensional Path Keeping at u = 1.5 ft/s

CTE steering controllers performed satisfactorily in conjunction with the diving controller.

The new concept of three dimensional path keeping was introduced. Development of the concept yields a control package that allows the AUV to follow a straight path line in three dimensional space.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The following are the major conclusions that can be drawn from this study:

- MIMO sliding mode controllers yield superior performance and enhanced capabilities as compared to SISO sliding mode controllers.
- The LQR method for MIMO sliding mode control design is preferred over pole placement because it allows the designer to choose which states to minimize resulting in specific desired control action. However, even the LQR method requires some experimentation to determine the best minimization matrix for the situation. Hence, this is an impetus for prior computer simulation.
- As first proposed by Lienard [Ref. 3], the speed, diving, and steering controllers can be designed separately - greatly simplifying the design process - and effectively control the vehicle simultaneously.
- The multi-level diving controller based on control efficiency at different speeds proved that it can effectively command the vehicle over the entire range of operational speeds.
- Both the LOS and CTE steering controllers did well in controlling the vehicle. They can be easily modified to compensate for observed underwater currents.
- Two coupled CTE controllers can command the vehicle to follow straight line paths in three dimensional space; however, consideration must be given to the mission planner so the ordered path is achievable. The vehicle is unable to follow a path line that is too steep.

B. RECOMMENDATIONS

The following actions should be taken to fully utilize the research done in this thesis:

- Design Kalman filters to reduce noise in the state measurements.
- Develop a more accurate and detailed model for the tunnel thrusters.
- Add horizontal tunnel thrusters to the model and design a multi-level steering controller, as was done for the diving controller.
- Develop software for switching of dive planes and rudders to ensure smooth transition between different way-points during three dimensional path keeping.

APPENDIX A. SUMMARY LISTING OF COMPUTER PROGRAMS

The following is a complete listing of all computer programs written and used in this thesis in the order they were discussed. The four programs contained in Appendices B to E are included because they are in the most general form for the particular application. None of the DISSPLA plotting programs are included in the Appendices because they are device dependent, but they are summarized below. All Fortran simulation programs utilize IMSL subroutines for matrix inversion and matrix/vector multiplication. Copies of any program used in this thesis can be obtained from:

Professor F. A. Papoulias, Code MEPa
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93943-5000

The listing follows:

SDVDIVLIN_I.FOR

Fortran program that solves the simplified nonlinear heave and pitch equations (with planes and vertical thrusters) for $\dot{w}$ and $\dot{q}$ at a particular surge velocity u.

SDVDIVPPTTPSF_POLE.FOR

Fortran program that simulates the AUV at all speeds (with planes and thrusters) in the dive plane only, via the simplified nonlinear equations of motion. Control laws determined by pole placement. Perfect state feedback.

SDVDIVPPTTPSF_LQR.FOR

Identical to SDVDIVPPTTPSF_POLE.FOR except the control laws are determined by the LQR method.

DPLOT2.FOR

Fortran/DISSPLA plotting program that produces graphs from the output of SDVDIVPPTTPSF_POLE.FOR and SDVDIVPPTTPSF_LQR.FOR.

SDVDIVPPTTOBS_LQR.FOR

Identical to SDVDIVPPTTPSF_LQR.FOR except it has a reduced order observer for heave velocity w.

DPLOT4.FOR

Fortran/DISSPLA plotting program that produces graphs from the output of SDVDIVPPTTOBS_LQR.FOR.

SDVSTRLIN.FOR

Fortran program that solves the simplified nonlinear sway and yaw equations (with two rudders) for $\dot{v}$ and $\dot{r}$ at a particular surge velocity u.

SDVLOSRR300PSF_LQR.FOR

Fortran program that simulates the AUV at high speeds (with two rudders) in the horizontal plane only, via the simplified nonlinear equations of motion. Control laws determined by the LQR method. State space representation determined from linearization at u = 3.00 ft/s. Perfect state feedback. Compensated for steady-state disturbances.

PLOT7A.FOR

Fortran/DISSPLA plotting program that produces graphs from the output of SDVLOSRR300PSF_LQR.FOR.

SDVLOSRR300OBS_LQR.FOR

Identical to SDVLOSRR300PSF_LQR.FOR except it has a reduced order observer for v, u_c, and v_c.

PLOT8.FOR

Fortran/DISSPLA plotting program that produces graphs from the output of SDVLOSRR300OBS_LQR.FOR.

SDVCTERR300OBS_LQR.FOR

Fortran program that simulates the AUV with CTE steering guidance at high speeds (with two rudders) in the horizontal plane only, via the simplified nonlinear equations of motion. Control laws determined by the LQR method. State space representation determined from linearization at u = 3.00 ft/s. Reduced order observer for v, u_c, and v_c. Compensated for steady-state disturbances by heading error.

SDVCTELOSRR300OBS_LQR.FOR

Identical to program SDVCTERR300OBS_LQR.FOR except that steady-state disturbances are compensated for by rudder action.

PLOT10.FOR

Fortran/DISSPLA program that produces graphs from the output of SDVCTERR300OBS_LQR.FOR and SDVCTELOSRR300OBS_LQR.FOR.

SDV3D_LOS.FOR

Fortran program that simulates the AUV with the full nonlinear equations of motion at high speeds only (two rudders/two planes). Combined controller containing the diving and LOS steering control laws. Perfect state feedback. Compensated for steady-state disturbances.

PLOT3D_LOS.FOR

Fortran/DISSPLA program that produces graphs from the output of SDV3D_LOS.FOR.

SDV3D_CTE.FOR

Identical to program SDV3D_LOS.FOR except the combined controller contains the CTE steering control laws.

PLOT3D_CTE.FOR

Fortran/DISSPLA program that produces graphs from the output of SDV3D_CTE.FOR.

APPENDIX B. MATRIX-x LQR DESIGN PROGRAM FOR MIMO SLIDING MODE CONTROLLERS

```
A=[-0.02270806,   -0.05053369,  2.5100E-2, 0.0;
     0.00562727,   -0.06273627, -6.6760E-2, 0.0;
     0.0,          1.0,         0.0,        0.0;
     1.0,          0.0,        -0.375,      0.0];
B=[  0.00017068, 0.00013609;
     .00004859, 0.00004343;
     0.0,       0.0;
     0.0,       0.0];
A11=A([1 2],[1 2]);
A12=A([1 2],[3 4]);
A21=A([3 4],[1 2]);
A22=A([3 4],[3 4]);
B1=B([1 2],[1 2]);
INQUIRE Q11
INQUIRE Q22
INQUIRE Q33
INQUIRE Q44
Q=DIAGONAL([Q11 Q22 Q33 Q44]);
Q11=Q([1 2],[1 2]);
Q12=Q([1 2],[3 4]);
Q21=Q([3 4],[1 2]);
Q22=Q([3 4],[3 4]);
QSTAR=Q22-Q21*INV(Q11)*Q12;
ASTAR=A22-A21*INV(Q11)*Q12;
[EVAL,KR,P]=REGULATOR(ASTAR,A21,QSTAR,Q11);
C2PRIME=INV(Q11)*(Q12+A21'*P);
C1PRIME=EYE(2);
CPRIME=[C1PRIME,C2PRIME];
Y1GAIN=-INV(B1)*(A11+C2PRIME*A21);
Y2GAIN=-INV(B1)*(A12+C2PRIME*A22);
YGAIN=[Y1GAIN,Y2GAIN];
XGAIN=YGAIN
SPRIME=CPRIME
POLES=EVAL
NEGINVB1=-INV(B1)
```

APPENDIX C. AUV DIVING SIMULATION PROGRAM; MIMO SLIDING MODE CONTROLLER DESIGNED BY LQR METHOD; *W* OBSERVED

```
      PROGRAM SDVDIVPPTTOBS_LQR
C*************************************************************************
C
C     PROGRAM SDVDIVPPTTOBS_LQR.FOR
C
C     WRITTEN BY LT TODD D. HAWKINSON, USN
C
C     FOR AUV THESIS WORK
C
C     THIS PROGRAM CONTROLS A 2-PLANE, 2-THRUSTER SDV IN THE
C     VERTICAL PLANE WITH SLIDING MODE CONTROL UTILIZING PLANES
C     AND THRUSTERS.
C
C     SPEED IS ALSO CONTROLLED USING SLIDING MODE CONTROLLER
C     DEVELOPED BY LIENARD.
C
C     THIS PROGRAM OBSERVES HEAVE VELOCITY, W.
C
C     THE OUTPUT OF THIS PROGRAM IS WRITTEN TO FILE, ALLOUT.DAT.
C     THIS FILE IS THEN ACCESSED BY PROGRAM, DPLOT4.FOR, TO GENERATE
C     OUTPUT GRAPHICS.
C
C     THIS PROGRAM UTITLIZES SUBROUTINES FROM THE IMSL MATH
C     LIBRARY, VERSION 1.1, (COPYRIGHT JANUARY 1989 BY IMSL, INC.).
C
C*************************************************************************
C
C  HEAVE CHARACTERISTICS
C
      REAL ZQ,ZQDOT,ZW,ZWDOT,ZDB,ZDS,CDZ,Z.ZDOT,W,WDOT
C
C  PITCH CHARACTERISTICS
C
      REAL MQ,MQDOT,MW,MWDOT,MDB,MDS,THE,THEDOT,Q,QDOT
C
C  VEHICLE CHARACTERISTICS
C
      REAL BOY,WT,L,XB,XG,ZB,ZG,IY,RHO,G,M,X(18),B(18),NU
C
C  SURGE CHARACTERISTICS
C
```

```
      REAL U,UDOT,UMAX,UCF,UD,N,RE,CD0,XUDOT,XWDB,
     &     XWDS,XQDB,XQDS,XWW,XQQ,XWQ,XDBDB,XDSDS,SIGNU
C
C PLANE CHARACTERISTICS
C
      REAL DMAX,DB,DS
C
C THRUSTER CHARACTERISTICS
C
      REAL IBV,ISV,IVMAX,XBV,XSV
C
C SLIDING MODE CONTROLLER VARIABLES
C
      REAL SI,EITA,S1,S2,S3,SAT1,SAT2,SAT3
C
C OBSERVER VARIABLES
C
      REAL Z1,WHAT
      COMMON Z1
C
C PROGRAM VARIABLES
C
      REAL MM(3,3),MMINV(3,3),F(3)
      REAL DELT,PIE
      REAL SGN
      REAL VEC1(18),VEC2(18)
      REAL DRAGDX,DRAGXDX
      REAL TIME,ZD
      INTEGER I,K,ITER,ISCREEN,IOUT
C
C HEAVE HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(ZQ=  -1.350E-1, ZQDOT=-6.810E-3, ZW=  -3.020E-1,
     &          ZWDOT=-2.430E-1, ZDB= -2.580E-2, ZDS= -7.255E-2,
     &          CDZ=  1.0)
C
C PITCH HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(MQ=  -6.860E-2, MQDOT=-1.680E-2, MW=  9.860E-2,
     &          MWDOT=-6.810E-3, MDB=  6.940E-3, MDS= -4.120E-2)
C
C VEHICLE CHARACTERISTICS
C
      PARAMETER(BOY=12000., WT=12000., L=17.425, XB=0.0, XG=0.0,
     &          ZB=0.0,   ZG=0.20,  IY=10000., RHO=1.940, G=32.2.
     &          NU=8.47E-4)
C
C SURGE CHARACTERISTICS
C
      PARAMETER(UMAX= 6.0, XUDOT=-7.580E-3, XWDB= 9.660E-3,
```

```
     &              XWDS= 4.600E-2,  XQDB=-2.600E-3,  XQDS= 2.610E-2,
     &              XWW= 1.710E-1,   XQQ=-1.470E-2,   XWQ=-1.920E-1,
     &              XDBDB=-8.070E-3, XDSDS=-1.160E-2)
C
C  PLANE CHARACTERISTICS
C
      PARAMETER(DMAX=0.4)
C
C  THRUSTER CHARACTERISTICS
C
      PARAMETER(IVMAX=20.0, XBV=6.70, XSV=6.70)
C
C  PROGRAM VARIABLES
C
      PARAMETER(DELT=0.01,PIE=3.141592654,ITER=20000)
C
C  DEFINE LENGTH X AND BEAM B TERMS FOR THE INTEGRATION
C
      X(1)=-105.9/12.
      X(2)=-104.3/12.
      X(3)=-99.3/12.
      X(4)=-94.3/12.
      X(5)=-87.3/12.
      X(6)=-76.8/12.
      X(7)=-66.3/12.
      X(8)=-55.8/12.
      X(9)=72.7/12.
      X(10)=79.2/12.
      X(11)=83.2/12.
      X(12)=87.2/12.
      X(13)=91.2/12.
      X(14)=95.2/12.
      X(15)=99.2/12.
      X(16)=101.2/12.
      X(17)=102.1/12.
      X(18)=103.2/12.
      B(1)=75.70/12.
      B(2)=75.70/12.
      B(3)=75.70/12.
      B(4)=75.70/12.
      B(5)=75.70/12.
      B(6)=75.70/12.
      B(7)=75.70/12.
      B(8)=75.70/12.
      B(9)=75.70/12.
      B(10)=70.24/12.
      B(11)=64.40/12.
      B(12)=58.16/12.
      B(13)=49.84/12.
      B(14)=38.88/12.
```

```
      B(15)=26.32/12.
      B(16)=18.56/12.
      B(17)=12.64/12.
      B(18)=0.00/12.
C
C  DATA ENTRY
C
      WRITE(*,*)
10    WRITE(*,*) 'ENTER ORDERED SPEED IN FT/SEC, UD = '
      READ(*,*) UD
      IF (UD .LT. 0.0 .OR. UD .GT. UMAX) THEN
            WRITE(*,*)
            WRITE(*,*) 'UD MUST BE:  0.0 < UD < 6.0'
            WRITE(*,*) 'RE-ENTER REALISTIC SPEED.'
            WRITE(*,*)
            GO TO 10
      ENDIF
20    WRITE(*,*) 'ENTER ORDERED DEPTH IN FEET, ZD = '
      READ(*,*) ZD
      IF (ZD .LT. 0.0) THEN
            WRITE(*,*)
            WRITE(*,*) 'ZD MUST BE:  ZD > 0.0'
            WRITE(*,*) 'RE-ENTER REALISTIC DEPTH'
            WRITE(*,*)
            GO TO 20
      ENDIF
      WRITE(*,*) 'ENTER SI ='
      READ(*,*) SI
40    WRITE(*,*) 'ENTER EITA = '
      READ(*,*) EITA
      IF (EITA .LE. 0.0) THEN
            WRITE(*,*)
            WRITE(*,*) 'EITA MUST BE: EITA > 0.0'
            WRITE(*,*) 'RE-ENTER REALISTIC EITA'
            WRITE(*,*)
            GO TO 40
      ENDIF
C
C  CALCULATE THE MASS
C
      M=WT/G
C
C  SET INITIAL VALUES
C
      U=UD
      N=UD/0.012
      DB=0.0
      DS=0.0
      IBV=0.0
      ISV=0.0
```

```
      W=0.0
      WHAT=0.0
      Z1=0.0
      Q=0.0
      THE=0.0
      Z=0.0
      WDOT=0.0
      QDOT=0.0
      UDOT=0.0
      THEDOT=0.0
      ZDOT=0.0
      TIME=0.0
C
C  CALCULATE THE MASS MATRIX
C
      MM(1,1)=M-(RHO/2.)*L*L*L*ZWDOT
      MM(1,2)=-(M*XG+(RHO/2.)*L*L*L*L*ZQDOT)
      MM(1,3)=0.0
      MM(2,1)=-(M*XG+(RHO/2.)*L*L*L*L*MWDOT)
      MM(2,2)=IY-(RHO/2.)*L*L*L*L*L*MQDOT
      MM(2,3)=M*ZG
      MM(3,1)=0.0
      MM(3,2)=M*ZG
      MM(3,3)=M-0.5*RHO*L*L*L*XUDOT
C
C  INVERT THE MASS MATRIX
C
      CALL LINRG(3,MM,3,MMINV,3)
C
C  INITIALIZE COUNTERS
C
      ISCREEN=1
      IOUT=1
C
C  OPEN OUTPUT FILE
C
      OPEN(UNIT=15,FILE='ALLOUT.DAT',STATUS='NEW')
      WRITE(15,*) TIME,ZL,Z,DB,DS,IBV,ISV,W,WHAT,Q,THE*180./PIE,UD,U,N
C
C  SIMULATION BEGINS
C
      DO 1000 I=1,ITER
C
C           CALCULATE THE DRAG FORCES
C
            DO 50 K=1,18
                  UCF=(W-Q*X(K))
                  SGN=1.0
                  IF (UCF .LT. 0.0) SGN=-1.0
                  VEC1(K)=B(K)*UCF*UCF*SGN
```

```
            VEC2(K)=B(K)*UCF*UCF*SGN*X(K)
50          CONTINUE
            CALL TRAP(18,VEC1,X,DRAGDX)
            CALL TRAP(18,VEC2,X,DRAGXDX)
C
C           RHS OF HEAVE EQUATION
C
            F(1) = (0.5*RHO*L*L*ZW*U)*W + (M*U+0.5*RHO*L*L*L*ZQ*U)*Q
     &           +(M*ZG)*Q*Q - (0.5*RHO*CDZ)*DRAGDX
     &           +(WT-BOY)*COS(THE) + (0.5*RHO*L*L*U*U*ZDB)*DB
     &           +(0.5*RHO*L*L*U*U*ZDS)*DS
     &           +(0.25-0.00833333*U)*IBV
     &           +(0.25-0.00833333*U)*ISV
C
C           RHS OF PITCH EQUATION
C
            F(2) = (0.5*RHO*L*L*L*MW*U)*W
     &           +(0.5*RHO*L*L*L*L*MQ*U-M*XG*U)*Q - (M*ZG)*W*Q
     &           +(0.5*RHO*CDZ)*DRAGXDX - (XG*WT-XB*BOY)*COS(THE)
     &           -(ZG*WT-ZB*BOY)*SIN(THE)
     &           +(0.5*RHO*L*L*L*U*U*MDB)*DB
     &           +(0.5*RHO*L*L*L*U*U*MDS)*DS
     &           -XBV*(0.25-0.00833333*U)*IBV
     &           +XSV*(0.25-0.00833333*U)*ISV
C
C           RHS OF SURGE EQUATION
C
            SIGNU=1.0
            IF ( (U .LT. 0.0) .AND. (N .GT. 0.0) ) SIGNU=-1.0
            IF ( (U .GT. 0.0) .AND. (N .LT. 0.0) ) SIGNU=-1.0
            RE=U*L/NU
            CD0=0.00385 + 1.296E-17*(RE-1.2E7)*(RE-1.2E7)
            F(3) = 0.5*RHO*L*L*U*W*(XWDB*DB+XWDS*DS)
     &           +0.5*RHO*L*L*L*U*Q*(XQDB*DB+XQDS*DS)
     &           +0.5*RHO*L*L*XWW*W*W
     &           +(M*XG+0.5*RHO*L*L*L*L*XQQ)*Q*Q
     &           +(0.5*RHO*L*L*L*XWQ-M)*W*Q
     &           +0.5*RHO*L*L*U*U*(XDBDB*DB*DB+XDSDS*DS*DS)
     &           -(WT-BOY)*SIN(THE)
     &           +0.5*RHO*L*L*CD0*(SIGNU*0.012*0.012*N*N-U*U)
            WDOT = MMINV(1,1)*F(1)+MMINV(1,2)*F(2)+MMINV(1,3)*F(3)
            QDOT = MMINV(2,1)*F(1)+MMINV(2,2)*F(2)+MMINV(2,3)*F(3)
            UDOT = MMINV(3,1)*F(1)+MMINV(3,2)*F(2)+MMINV(3,3)*F(3)
            THEDOT = Q
            ZDOT = -U*SIN(THE) + W*COS(THE)
C
C           FIRST ORDER INTEGRATION
C
            W = W + DELT*WDOT
            Q = Q + DELT*QDOT
```

```
      U = U + DELT*UDOT
      THE = THE + DELT*THEDOT
      Z = Z + DELT*ZDOT
      IF (Z .LT. 0.0) Z=0.0
C
C     CALCULATE THE CONTROL LAW
C
      IF (U .LE. 0.75) THEN
            DB=0.0
            DS=0.0
            CALL OBSERV1(I,U,Q,THE,Z,ZD,IBV,ISV,DELT,WHAT)
            S1=1.0*WHAT + 0.0*Q - 0.1452*THE + 0.4471*(Z-ZD)
            S2=0.0*WHAT + 1.0*Q + 0.7074*THE - 0.0007*(Z-ZD)
            IF (ABS(S1) .LT. SI) SAT1=S1/SI
            IF (S1 .LE. -SI) SAT1=-1.0
            IF (S1 .GE.  SI) SAT1=1.0
            IF (ABS(S2) .LT. SI) SAT2=S2/SI
            IF (S2 .LE. -SI) SAT2=-1.0
            IF (S2 .GE. SI) SAT2=1.0
            IBV = -1.2666E3*WHAT + 6.8616E3*Q - 0.2037E3*THE
     &           -0.3097E4*EITA*SAT1 + 0.9703E4*EITA*SAT2
            ISV = -1.5299E3*WHAT - 7.1672E3*Q + 1.3030E3*THE
     &           -0.3464E4*EITA*SAT1 - 1.2169E4*EITA*SAT2
      ELSEIF ( (U .GT. 0.75) .AND. (U .LE. 1.75)) THEN
            CALL OBSERV2(I,U,Q,THE,Z,ZD,DB,IBV,DELT,WHAT)
            S1=1.0*WHAT + 0.0*Q - 0.0735*THE + 0.0678*(Z-ZD)
            S2=0.0*WHAT + 1.0*Q + 1.0855*THE - 0.0735*(Z-ZD)
            IF (ABS(S1) .LT. SI) SAT1=S1/SI
            IF (S1 .LE. -SI) SAT1=-1.0
            IF (S1 .GE.  SI) SAT1=1.0
            IF (ABS(S2) .LT. SI) SAT2=S2/SI
            IF (S2 .LE. -SI) SAT2=-1.0
            IF (S2 .GE. SI) SAT2=1.0
            DB  =  0.0051E3*WHAT - 0.0818E3*Q - 0.0023E3*THE
     &           -0.0016E3*EITA*SAT1 - 0.0937E3*EITA*SAT2
            IBV = -0.1347E3*WHAT + 3.3747E3*Q + 0.2712E3*THE
     &           -3.3095E3*EITA*SAT1 + 2.9370E3*EITA*SAT2
            DS=-DB
            IF (U .LT. 1.25) DS=0.0
            ISV=IBV
            IF (U .GT. 1.25) ISV=0.0
      ELSE
            IBV=0.0
            ISV=0.0
            CALL OBSERV3(I,U,Q,THE,Z,ZD,DB,DS,DELT,WHAT)
            S1=1.0*WHAT + 0.0*Q - 0.0945*THE + 0.0328*(Z-ZD)
            S2=0.0*WHAT + 1.0*Q + 1.3127*THE - 0.0945*(Z-ZD)
            IF (ABS(S1) .LT. SI) SAT1=S1/SI
            IF (S1 .LE. -SI) SAT1=-1.0
            IF (S1 .GE.  SI) SAT1=1.0
```

```
                IF (ABS(S2) .LT. SI) SAT2=S2/SI
                IF (S2 .LE. -SI) SAT2=-1.0
                IF (S2 .GE. SI) SAT2=1.0
                DB = -1.2124*WHAT - 17.7554*Q - 6.3692*THE
     &              +9.2671*EITA*SAT1 - 18.1220*EITA*SAT2
                DS = -0.6568*WHAT +  4.4622*Q +  2.3181*THE
     &              +1.7528*EITA*SAT1 + 8.3426*EITA*SAT2
            ENDIF
C
C           SPEED CONTROLLER
C
            S3=U-UD
            IF (ABS(S3) .LT. SI) SAT3=S3/SI
            IF (S3 .LE. -SI) SAT3=-1.0
            IF (S3 .GE.  SI) SAT3=1.0
            N=-1153.9*SAT3 + 83.33*U
C
C           CHECK FOR SATURATION
C
            IF (DB .GT. DMAX) DB=DMAX
            IF (DB .LT. -DMAX) DB=-DMAX
            IF (DS .GT. DMAX) DS=DMAX
            IF (DS .LT. -DMAX) DS=-DMAX
            IF (IBV .GT. IVMAX) IBV=IVMAX
            IF (IBV .LT. -IVMAX) IBV=-IVMAX
            IF (ISV .GT. IVMAX) ISV=IVMAX
            IF (ISV .LT. -IVMAX) ISV=-IVMAX
            IF (N .GT. 500.) N=500.
            IF (N .LT. -500.) N=-500.
C
C           PRINT TO SCREEN
C
            IF (ISCREEN .EQ. 100) THEN
                TIME=I/100.
                WRITE(*,900) TIME,ZD,Z,W,WHAT
900             FORMAT(' TIME= ',F6.2,1X,'ZD= ',F6.2,1X,'Z= ',
     &                  F8.4,1X,'W= ',F7.4,1X,'WHAT= ',F14.7)
                ISCREEN=0
            ENDIF
C
C           PRINT TO FILE
C
            IF (IOUT .EQ. 10) THEN
                TIME=I/100.
                WRITE(15,*) TIME,ZD,Z,DB,DS,IBV,ISV,W,WHAT,Q,
     &                      THE*180./PIE,UD,U,N
                IOUT=0
            ENDIF
            ISCREEN=ISCREEN+1
            IOUT=IOUT+1
```

```
1000  CONTINUE
      CLOSE(UNIT=15)
      STOP
      END
C
C  NUMERICAL INTEGRATION ROUTINE USING THE TRAPEZOIDAL RULE
C
      SUBROUTINE TRAP(N,A,B,OUT)
      INTEGER N,N1,I
      REAL A(N),B(N),OUT,OUT1
      N1=N-1
      OUT=0.0
      DO 10 I=1,N1
            OUT1 = 0.5*(A(I) + A(I+1))  *  (B(I+1) - B(I))
            OUT = OUT + OUT1
10    CONTINUE
      RETURN
      END
C
C  OBSERVER FOR THE SPEED RANGE:  0.00 < U < 0.75
C
      SUBROUTINE OBSERV1(I,U,Q,THE,Z,ZD,IBV,ISV,DELT,WHAT)
      INTEGER I
      REAL U,Q,THE,Z,ZD,IBV,ISV,DELT,WHAT
      REAL Z1,Z1DOT
      COMMON Z1
      REAL L1,L2,L3
      REAL F
      REAL G1,G2,G3
      REAL H1,H2
      REAL A11,A12,A13,B11,B12,A21,A22,A23,B21,B22,A32,A41,A43
      REAL S1
      PARAMETER(S1=-0.9)
      A11=-0.060554828*U
      A12=-0.134756503*U
      A13= 0.02510025
      B11= 6.913620E-04*(0.25000000-8.33333333E-03*U)
      B12= 5.512506E-04*(0.25000000-8.33333333E-03*U)
      A21= 0.015006049*U
      A22=-0.167296702*U
      A23=-0.06676024
      B21=-1.968203E-04*(0.25000000-8.33333333E-03*U)
      B22= 1.759190E-04*(0.25000000-8.33333333E-03*U)
      A32= 1.0
      A41= 1.0
      A43=-U
      L1=(A11-S1)/(A21+A41)
      L2=0.0
      L3=L1
      F=A11-L1*A21-L3*A41
```

```
      H1=B11-L1*B21
      H2=B12-L1*B22
      G1=A12-L1*A22-L2*A32+F*L1
      G2=A13-L1*A23-L3*A43+F*L2
      G3=F*L3
      Z1DOT=F*Z1+G1*Q+G2*THE+G3*(Z-ZD)+H1*IBV+H2*ISV
      Z1=Z1+Z1DOT*DELT
      WHAT=L1*Q+L2*THE+L3*(Z-ZD)+Z1
      IF (I .LE. 1000) WHAT=0.0
      RETURN
      END
C
C  OBSERVER FOR THE SPEED RANGE:  0.75 < U < 1.75
C
      SUBROUTINE OBSERV2(I,U,Q,THE,Z,ZD,DB,IBV,DELT,WHAT)
      INTEGER I
      REAL U,Q,THE,Z,ZD,DB,IBV,DELT,WHAT
      REAL Z1,Z1DOT
      COMMON Z1
      REAL L1,L2,L3
      REAL F
      REAL G1,G2,G3
      REAL H1,H2
      REAL A11,A12,A13,B11,B12,A21,A22,A23,B21,B22,A32,A41,A43
      REAL S1
      PARAMETER(S1=-0.32)
      A11=-0.060554828*U
      A12=-0.134756503*U
      A13= 0.02510025
      B11=-0.005093606*U*U
      B12= 6.913620E-04*(0.25000000-8.33333333E-03*U)
      A21= 0.015006049*U
      A22=-0.167296702*U
      A23=-0.06676024
      B21= 0.001070202*U*U
      B22=-1.968203E-04*(0.25000000-8.33333333E-03*U)
      A32= 1.0
      A41= 1.0
      A43=-U
      L1=(A11-S1)/(A21+A41)
      L2=0.0
      L3=L1
      F=A11-L1*A21-L3*A41
      H1=B11-L1*B21
      H2=B12-L1*B22
      G1=A12-L1*A22-L2*A32+F*L1
      G2=A13-L1*A23-L3*A43+F*L2
      G3=F*L3
      Z1DOT=F*Z1+G1*Q+G2*THE+G3*(Z-ZD)+H1*DB+H2*IBV
      Z1=Z1+Z1DOT*DELT
```

```
      WHAT=L1*Q+L2*THE+L3*(Z-ZD)+Z1
      IF (I .LE. 1000) WHAT=0.0
      RETURN
      END
C
C  OBSERVER FOR THE SPEED RANGE:  U > 1.75
C
      SUBROUTINE OBSERV3(I,U,Q,THE,Z,ZD,DB,DS,DELT,WHAT)
      INTEGER I
      REAL U,Q,THE,Z,ZD,DB,DS,DELT,WHAT
      REAL Z1,Z1DOT
      COMMON Z1
      REAL L1,L2,L3
      REAL F
      REAL G1,G2,G3
      REAL H1,H2
      REAL A11,A12,A13,B11,B12,A21,A22,A23,B21,B22,A32,A41,A43
      REAL S1
      PARAMETER(S1=-0.88)
      A11=-0.060554828*U
      A12=-0.134756503*U
      A13= 0.02510025
      B11=-0.005093606*U*U
      B12=-0.011064514*U*U
      A21= 0.015006049*U
      A22=-0.167296702*U
      A23=-0.06676024
      B21= 0.001070202*U*U
      B22=-0.005658107*U*U
      A32= 1.0
      A41= 1.0
      A43=-U
      L1=(A11-S1)/(A21+A41)
      L2=0.0
      L3=L1
      F=A11-L1*A21-L3*A41
      H1=B11-L1*B21
      H2=B12-L1*B22
      G1=A12-L1*A22-L2*A32+F*L1
      G2=A13-L1*A23-L3*A43+F*L2
      G3=F*L3
      Z1DOT=F*Z1+G1*Q+G2*THE+G3*(Z-ZD)+H1*DB+H2*DS
      Z1=Z1+Z1DOT*DELT
      WHAT=L1*Q+L2*THE+L3*(Z-ZD)+Z1
      IF (I .LE. 1000) WHAT=0.0
      RETURN
      END
```

APPENDIX D. AUV STEERING SIMULATION PROGRAM; MIMO SLIDING MODE LOS CONTROLLER DESIGNED BY LQR METHOD; DISTURBANCE REJECTION; V, U_C, AND V_C OBSERVED

```
      PROGRAM SDVLOSRR300OBS_LQR
C*************************************************************************
C
C     PROGRAM SDVLOSRR300OBS_LQR.FOR
C
C     WRITTEN BY LT TODD D. HAWKINSON, USN
C
C     FOR AUV THESIS WORK
C
C     THIS PROGRAM CONTROLS A 2 RUDDER SDV WITH A SLIDING MODE
C     LINE-OF-SIGHT CONTROLLER FOR STEERING.
C
C     SPEED IS ALSO CONTROLLED USING A SLIDING MODE CONTROLLER
C     DEVELOPED BY LIENARD.
C
C     THE EQUATIONS HAVE BEEN LINEARIZED AT U = 3.0 FT/S TO OBTAIN
C     THE SLIDING MODE CONTROL LAWS.
C
C     THE PROGRAM HAS OBSERVERS FOR V, VC, AND UC.
C
C     THE OUTPUT OF THE PROGRAM IS WRITTEN TO FILES, LOSCURR.DAT,
C     LOSWAYPT.DAT, LOSALLOUT.DAT.  THESE FILES ARE THEN ACCESSED
C     BY THE DISSPLA PLOTTING PROGRAM, PLOT8.FOR, TO GENERATE
C     OUTPUT GRAPHICS.
C
C     THIS PROGRAM UTILIZES SUBROUTINES FROM THE IMSL MATH
C     LIBRARY, VERSION 1.1, (COPYRIGHT JANUARY 1989 BY IMSL, INC.).
C
C*************************************************************************
C
C  YAW CHARACTERISTICS
C
      REAL NR,NRDOT,NV,NVDOT,NDBR,NDSR,CDY,R,RDOT,PSI,PSIDOT,
     &     PSIDX,PSIX
C
C  SWAY CHARACTERISTICS
C
      REAL YR,YRDOT,YV,YVDOT,V,VDOT,YDBR,YDSR
C
C  SURGE CHARACTERISTICS
C
```

```
      REAL U,UMAX,UD,N,RE,ETA,CD0,XPROP,XUDOT,XVR,XRR,XRDBR,
     &     XRDSR,XVV,XVDBR,XVDSR,XDBRDBR,XDSRDSR,SIGNU,UDOT
C
C  RUDDER CHARACTERISTICS
C
      REAL DMAX,DBR,DSR
C
C  SLIDING MODE CONTROL VARIABLES
C
      REAL SI,EITA,SS1,SS2,SAT1,SAT2,SS3,SAT3
C
C  VEHICLE CHARACTERISTICS
C
      REAL WT,L,XG,YG,IZ,RHO,G,M,X(18),H(18),NU
C
C  NAVIGATOR VARIABLES
C
      REAL UCO,UC,VCO,VC
      REAL TARGET,PSID,DAWAY,ALPHA
      REAL XD,XD1,XD2,XPOS,XCURR,XCTE,YD,YD1,YD2,YPOS,YCURR,YCTE
      REAL XDOT,YDOT
      INTEGER IWAY,INAV
C
C  OBSERVER VARIABLES
C
      REAL A11,A12,A21,A22,B11,B12,B21,B22,L12,L23,L34,VHAT,UCHAT,
     &     VCHAT,UCOHAT,VCOHAT,Z1,Z2,Z3,Z1DOT,Z2DOT,Z3DOT,S1,S2,S3
C
C  PROGRAM VARIABLES
C
      REAL MM(3,3),MMINV(3,3),F(3),OUT(3)
      REAL DELT
      REAL PIE
      REAL SGN
      REAL VEC1(18),VEC2(18)
      REAL DRAGDX,DRAGXDX,UCF
      REAL TIME
      INTEGER I,ITER,ISCREEN,K,IOUT
C
C  YAW HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(NR=  -1.640E-2, NRDOT=-3.400E-3, NV=  -7.420E-3,
     &          NVDOT= 1.240E-3, NDBR=  1.290E-2, NDSR= -1.290E-2,
     &          CDY=  3.500E-1)
C
C  SWAY HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(YR=   2.970E-2, YRDOT= 1.240E-3, YV=  -9.310E-2,
     &          YVDOT=-5.550E-2, YDBR=  2.730E-2, YDSR=  2.730E-2)
C
```

```
C SURGE HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(UMAX=   6.0    , XUDOT= -7.580E-3, XVR=  1.890E-2,
     &          XRR=   4.010E-3, XRDBR=  8.180E-4, XRDSR=-8.180E-4,
     &          XVV=   5.290E-2, XVDBR=  1.730E-3, XVDSR= 1.730E-3,
     &          XDBRDBR=-1.010E-2, XDSRDSR=-1.010E-2)
C
C RUDDER CHARACTERISTICS
C
      PARAMETER(DMAX=0.4)
C
C VEHICLE CHARACTERISTICS
C
      PARAMETER(WT=12000., L=17.425, XG=0.0, YG=0.0, IZ=10000.,
     &          RHO=1.940, G=32.2,   NU=8.47E-4)
C
C NAVIGATOR VARIABLES
C
      PARAMETER(TARGET=8.71250)
C
C OBSERVER VARIABLES
C
      PARAMETER(A11=-0.042174797, A12=-0.351578237,
     &          A21=-0.002794897, A22=-0.098415870,
     &          B11= 0.012974539, B12= 0.011513052,
     &          B21= 0.004421618, B22=-0.004244119,
     &          S1= -1.0,         S2= -1.1,
     &          S3= -1.2)
C
C PROGRAM VARIABLES
C
      PARAMETER(PIE=3.141593,DELT=0.01,ITER=50000)
C
C DEFINE LENGTH X AND HEIGHT TERMS FOR THE INTEGRATION
C
      X(1)=-105.9/12.
      X(2)=-104.3/12.
      X(3)=-99.3/12.
      X(4)=-94.3/12.
      X(5)=-87.3/12.
      X(6)=-76.8/12.
      X(7)=-66.3/12.
      X(8)=-55.8/12.
      X(9)=72.7/12.
      X(10)=79.2/12.
      X(11)=83.2/12.
      X(12)=87.2/12.
      X(13)=91.2/12.
      X(14)=95.2/12.
      X(15)=99.2/12.
```

```
      X(16)=101.2/12.
      X(17)=102.1/12.
      X(18)=103.2/12.
      H(1)=0.0/12.
      H(2)=2.28/12.
      H(3)=8.24/12.
      H(4)=13.96/12.
      H(5)=19.76/12.
      H(6)=25.10/12.
      H(7)=29.36/12.
      H(8)=31.85/12.
      H(9)=31.85/12.
      H(10)=30.00/12.
      H(11)=27.84/12.
      H(12)=25.12/12.
      H(13)=21.44/12.
      H(14)=17.12/12.
      H(15)=12.00/12.
      H(16)=9.12/12.
      H(17)=6.72/12.
      H(18)=0.00/12.
C
C  DATA ENTRY
C
      WRITE(*,*)
10    WRITE(*,*) 'ENTER ORDERED SPEED IN FT/SEC, UD = ?'
      READ(*,*) UD
      IF (UD .LT. 0.0 .OR. UD .GT. UMAX) THEN
            WRITE(*,*)
            WRITE(*,*) ' UD MUST BE:  0.0 <= UD <= 6.0'
            WRITE(*,*) ' RE-ENTER REALISTIC SPEED . . .'
            WRITE(*,*)
            GO TO 10
      ENDIF
      WRITE(*,*)
      WRITE(*,*) 'ENTER GLOBAL CURRENTS, UCO,VCO = ?'
      READ(*,*) UCO,VCO
      OPEN(UNIT=19,FILE='LOSCURR.DAT',STATUS='NEW')
      WRITE(19,25) UCO,VCO
25    FORMAT(F6.3,1X,F6.3)
      CLOSE(UNIT=19)
      WRITE(*,*)
30    WRITE(*,*) 'ENTER CONTROLLER SI ='
      READ(*,*) SI
      WRITE(*,*)
40    WRITE(*,*) 'ENTER CONTROLLER EITA = '
      READ(*,*) EITA
      IF (EITA .LE. 0.0) THEN
            WRITE(*,*)
            WRITE(*,*) 'EITA MUST BE: EITA > 0.0'
```

```
            \. RITE(*,*) 'RE-ENTER REALISTIC EITA'
            WRITE(*,*)
            GO TO 40
         ENDIF
         WRITE(*,*)
         WRITE(*,*) 'INITIAL HEADING IS 000 DEGREES (DUE NORTH)'
         WRITE(*,*) 'INITIAL POSITION IS (0.0,0.0)'
         WRITE(*,*) 'ENTER FIRST WAY-POINT:  X,Y = ?'
         READ(*,*) XD,YD
C
C  SET INITIAL STARTING POSITION
C
         XD1=0.0
         YD1=0.0
         XD2=XD
         YD2=YD
         OPEN(UNIT=20,FILE='LOSWAYPT.DAT',STATUS='NEW')
         WRITE(20,*) XD1,YD1
         WRITE(20,*) XD2,YD2
C
C  CALCULATE THE MASS
C
         M=WT/G
C
C  SET INITIAL VALUES
C
         ALPHA=ATAN2((YD2-YD1),(XD2-XD1))
         UC=UCO*COS(ALPHA)+VCO*SIN(ALPHA)
         VC=VCO*COS(ALPHA)-UCO*SIN(ALPHA)
         U=UD
         N=UD/0.012
         DBR=0.0
         DSR=0.0
         V=0.0
         R=0.0
         PSI=0.0
         PSID=0.0
         XPOS=0.0
         YPOS=0.0
         XCURR=XPOS
         YCURR=YPOS
         XCTE=0.0
         YCTE=0.0
         TIME=0.0
         VHAT=0.0
         UCHAT=0.0
         VCHAT=0.0
         UCOHAT=0.0
         VCOHAT=0.0
         Z1=0.0
```

```
      Z2=0.0
      Z3=0.0
      Z1DOT=0.0
      Z2DOT=0.0
      Z3DOT=0.0
C
C  CALCULATE THE MASS MATRIX
C
      MM(1,1)=(M-0.5*RHO*L*L*L*XUDOT)
      MM(1,2)=0.0
      MM(1,3)=-M*YG
      MM(2,1)=0.0
      MM(2,2)=(M-0.5*RHO*L*L*L*YVDOT)
      MM(2,3)=(M*XG-0.5*RHO*L*L*L*L*YRDOT)
      MM(3,1)=-M*YG
      MM(3,2)=(M*XG-0.5*RHO*L*L*L*L*NVDOT)
      MM(3,3)=(IZ-0.5*RHO*L*L*L*L*L*NRDOT)
C
C  INVERT THE MASS MATRIX USING IMSL LIBRARY SUBROUTINE
C
      CALL LINRG(3,MM,3,MMINV,3)
C
C  INITIALIZE THE COUNTERS
C
      ISCREEN=1
      INAV=0
      IWAY=1
      IOUT=1
C
C  OPEN OUTPUT FILES
C
      OPEN(UNIT=18,FILE='LOSALLOUT.DAT',STATUS='NEW')
      WRITE(18,*) TIME,XPOS,YPOS,PSID,PSI,DBR,DSR,UD,U,V,VHAT,R,
     &            UC,UCHAT,VC,VCHAT
C
C  SIMULATION BEGINS
C
      DO 1000 I=1,ITER
C
C          CALCULATE THE DRAG FORCES
C
          DO 50 K=1,18
              UCF=(V+R*X(K))
              SGN=1.0
              IF (UCF .LT. 0.0) SGN=-1.0
              VEC1(K)=H(K)*UCF*UCF*SGN
              VEC2(K)=H(K)*UCF*UCF*SGN*X(K)
50        CONTINUE
          CALL TRAP(18,VEC1,X,DRAGDX)
          CALL TRAP(18,VEC2,X,DRAGXDX)
```

```
C
C          RHS OF SURGE EQUATION
C
           SIGNU=1.0
           IF ( (U .LT. 0.0) .AND. (N .GT. 0.0) ) SIGNU=-1.0
           IF ( (U .GT. 0.0) .AND. (N .LT. 0.0) ) SIGNU=-1.0
           RE=U*L/NU
           CD0=0.00385 + 1.296E-17*(RE-1.2E7)*(RE-1.2E7)
           F(1)=0.5*RHO*L*L*U*V*(XVDBR*DBR + XVDSR*DSR)
     &       + 0.5*RHO*L*L*L*U*R*(XRDBR*DBR + XRDSR*DSR)
     &       + 0.5*RHO*L*L*XVV*V*V
     &       + (M*XG + 0.5*RHO*L*L*L*L*XRR)*R*R
     &       + (M + 0.5*RHO*L*L*L*XVR)*V*R
     &       + 0.5*RHO*L*L*U*U*(XDBRDBR*DBR*DBR + XDSRDSR*DSR*DSR)
     &       + 0.5*RHO*L*L*CD0*(SIGNU*0.012*0.012*N*N - U*U)
C
C          RHS OF SWAY EQUATION
C
           F(2)=0.5*RHO*L*L*YV*U*V
     &       + (0.5*RHO*L*L*L*YR - M)*U*R
     &       + M*YG*R*R
     &       + 0 5*RHO*L*L*U*U*(YDBR*DBR + YDSR*DSR)
     &       - 0.5*RHO*CDY*DRAGDX
C
C          RHS OF YAW EQUATION
C
           F(3)=0.5*RHO*L*L*L*NV*U*V
     &       + (0.5*RHO*L*L*L*L*NR*U - M*XG*U)*R
     &       - M*YG*V*R
     &       + 0.5*RHO*L*L*L*U*U*(NDBR*DBR + NDSR*DSR)
     &       - 0.5*RHO*CDY*DRAGXDX
C
C          DEFINITION
C
           PSIDOT=R
C
C          MULTIPLY INVERTED MASS MATRIX AND F VECTOR
C
           CALL MURRV(3,3,MMINV,3,3,F,1,3,OUT)
           UDOT=OUT(1)
           VDOT=OUT(2)
           RDOT=OUT(3)
           XDOT=UCO+U*COS(PSI)-V*SIN(PSI)
           YDOT=VCO+U*SIN(PSI)+V*COS(PSI)
C
C          FIRST ORDER INTEGRATION
C
           U=U + DELT*UDOT
           V=V + DELT*VDOT
           R=R + DELT*RDOT
```

```
      PSI=PSI + DELT*PSIDOT
C
C     MAKE PSI TO BE:  -180 < PSI <= 180 DEGREES
C
      IF (PSI .GT. PIE) THEN
            PSI=PSI - 2.0*PIE
      ELSEIF (PSI .LE. -PIE) THEN
            PSI=PSI + 2.0*PIE
      ENDIF
      XPOS=XPOS+DELT*XDOT
      YPOS=YPOS+DELT*YDOT
C
C     CHECK IF TIME FOR NAV UPDATE
C
      INAV=INAV+1
      IF (INAV .NE. 1) GO TO 80
      INAV=0
      XCURR=XPOS
      YCURR=YPOS
      DAWAY=SQRT((XCURR-XD)**2 + (YCURR-YD)**2)
C
C     CHECK DISTANCE TO WAY-POINT
C
      IF (DAWAY .LT. TARGET) THEN
C
C           MISS DISTANCE TO SCREEN, ENTER NEW WAY-POINT,
C           CALCULATE NEW ALPHA,UCHAT,VCHAT,XCTE,YCTE,Z2,Z3
C
            WRITE(*,*)
            WRITE(*,60) IWAY
60          FORMAT(' WAY-POINT #',I1,' REACHED.')
            WRITE(*,65) DAWAY
65          FORMAT(' DAWAY = ',F8.3,' FEET')
            UCOHAT=UCHAT*COS(ALPHA)-VCHAT*SIN(ALPHA)
            VCOHAT=VCHAT*COS(ALPHA)+UCHAT*SIN(ALPHA)
            IWAY=IWAY+1
            WRITE(*,70) IWAY
70          FORMAT(' ENTER WAY-POINT #',I1,':  X,Y = ?')
            READ(*,*) XD,YD
            XD1=XD2
            YD1=YD2
            XD2=XD
            YD2=YD
            WRITE(20,*) XD2,YD2
            ALPHA=ATAN2((YD2-YD1),(XD2-XD1))
            UC=UCO*COS(ALPHA)+VCO*SIN(ALPHA)
            VC=VCO*COS(ALPHA)-UCO*SIN(ALPHA)
            UCHAT=UCOHAT*COS(ALPHA)+VCOHAT*SIN(ALPHA)
            VCHAT=VCOHAT*COS(ALPHA)-UCOHAT*SIN(ALPHA)
            XCTE=(YCURR-YD1)*SIN(ALPHA)
```

```
     &                +(XCURR-XD1)*COS(ALPHA)
                  YCTE=(YCURR-YD1)*COS(ALPHA)
     &                -(XCURR-XD1)*SIN(ALPHA)
                  Z2=VCHAT+S2*YCTE
                  Z3=UCHAT+S3*XCTE+U*COS(PSI-ALPHA)
               ENDIF
C
C              CALCULATE NEW PSID DUE TO CHANGE OF XD,YD AND/OR
C              XCURR,YCURR
C
80             PSID=ATAN2((YD-YCURR),(XD-XCURR))
               XCTE=(YCURR-YD1)*SIN(ALPHA)+(XCURR-XD1)*COS(ALPHA)
               YCTE=(YCURR-YD1)*COS(ALPHA)-(XCURR-XD1)*SIN(ALPHA)
C
C              CALCULATE THE QUICKEST ROUTE PSID
C
               IF((PSI  .GE.  0.0 .AND. PSI  .LE. PIE) .AND.
     &            (PSID .GT. -PIE .AND. PSID .LT. PSI-PIE)) THEN
                  PSID=PSID + 2.0*PIE
               ELSEIF((PSI  .GT. -PIE   .AND. PSI  .LT. 0.0) .AND.
     &            (PSID .GT. PSI+PIE .AND. PSID .LE. PIE)) THEN
                  PSID=PSID - 2.0*PIE
               ENDIF
C*****************************************************************************
C              OBSERVERS FOR SWAY VELOCITY AND CURRENT VELOCITIES
C*****************************************************************************
               L12=(A11*U-S1)/(A21*U)
               L23=-S2
               L34=-S3
               Z1DOT=S1*Z1 + (A12*U-L12*A22*U+S1*L12)*R
     &                + (B11-B21*L12)*U*U*DBR + (B12-B22*L12)*U*U*DSR
               Z2DOT=S2*Z1 + S2*Z2 - L23*U*SIN(PSI-ALPHA) + S2*L12*R
     &                + S2*L23*YCTE
               Z3DOT=S3*Z3 - S3*S3*XCTE
C
C              FIRST ORDER INTEGRATION
C
               Z1=Z1+DELT*Z1DOT
               Z2=Z2+DELT*Z2DOT
               Z3=Z3+DELT*Z3DOT
C
C              FINAL OBSERVER EQUATIONS
C
               VHAT=L12*R + Z1
               VCHAT=L23*YCTE + Z2
               UCHAT=Z3 - S3*XCTE - U*COS(PSI-ALPHA)
C*****************************************************************************
C              END OBSERVERS
C*****************************************************************************
C
```

```
C       CALCULATE THE CONTROL LAWS
C
        IF (VC/U .GT. 1.0) THEN
            ARG=1.0
        ELSEIF (VC/U .LT. -1.0) THEN
            ARG=-1.0
        ELSE
            ARG=VC/U
        ENDIF
        SS1=1.0*V
        SS2=1.0*R + 0.5*(PSI-PSID) + 0.5*ASIN(ARG)
        SS3=U-UD
        IF (ABS(SS1) .LT. SI) SAT1=SS1/SI
        IF (SS1 .LE. -SI) SAT1=-1.0
        IF (SS1 .GE. SI) SAT1=1.0
        IF (ABS(SS2) .LT. SI) SAT2=SS2/SI
        IF (SS2 .LE. -SI) SAT2=-1.0
        IF (SS2 .GE. SI) SAT2=1.0
        IF (ABS(SS3) .LT. 1.0) SAT3=SS3/1.0
        IF (SS3 .LE. -1.0) SAT3=-1.0
        IF (SS3 .GE. 1.0) SAT3=1.0
        DBR= 0.6642*VHAT + 2.2219*R - EITA*4.4499*SAT1
     &       - EITA*12.0713*SAT2
        DSR= 0.4725*VHAT + 7.6752*R - EITA*4.6361*SAT1
     &       + EITA*13.6037*SAT2
        N=-1153.9*SAT3 + 83.33*U
C
C       CHECK FOR SATURATION
C
        IF (DBR .GT. DMAX) DBR=DMAX
        IF (DBR .LT. -DMAX) DBR=-DMAX
        IF (DSR .GT. DMAX) DSR=DMAX
        IF (DSR .LT. -DMAX) DSR=-DMAX
        IF (N .GE. 500.0) N=500.0
        IF (N .LE. -500.0) N=-500.0
C
C       OUTPUT
C
        IF (PSID .LT. 0.0 .AND. PSID .GT. -2.0*PIE) THEN
            PSIDX=(PSID+2.0*PIE)*180.0/PIE
        ELSE
            PSIDX=PSID*180.0/PIE
        ENDIF
        IF (PSI  .LT. 0.0 .AND. PSI  .GT. -PIE) THEN
            PSIX=(PSI+2.0*PIE)*180.0/PIE
        ELSE
            PSIX=PSI*180.0/PIE
        ENDIF
        IF (ISCREEN .EQ. 200) THEN
            WRITE(*,90) PSIDX,PSIX,XPOS,YPOS
```

```
90              FORMAT(' PSID=',F9.4,2X,'PSI=',F9.4,2X,
     &                  'XPOS= 'F8.2,2X,'YPOS= ',F8.2)
                ISCREEN=0
            ENDIF
            IF (IOUT .EQ. 20) THEN
                TIME=I/100.
                WRITE(18,*) TIME,XPOS,YPOS,PSIDX,PSIX,DBR,DSR,
     &                      UD,U,V,VHAT,R,UC,UCHAT,VC,VCHAT
                IOUT=0
            ENDIF
            IOUT=IOUT+1
            ISCREEN=ISCREEN+1
1000    CONTINUE
        CLOSE(UNIT=18)
        CLOSE(UNIT=20)
        STOP
        END
C
C       NUMERICAL INTEGRATION ROUTINE USING THE TRAPEZOIDAL RULE
C
        SUBROUTINE TRAP(N,A,B,OUT)
        INTEGER N,N1,I
        REAL A(N),B(N),OUT,OUT1
        N1=N-1
        OUT=0.0
        DO 10 I=1,N1
            OUT1 = 0.5*(A(I) + A(I+1))  *  (B(I+1) - B(I))
            OUT = OUT + OUT1
10      CONTINUE
        RETURN
        END
```

APPENDIX E. AUV STEERING SIMULATION PROGRAM; MIMO SLIDING MODE CTE CONTROLLER DESIGNED BY LQR METHOD; DISTURBANCE REJECTION; V, U_C, AND V_C OBSERVED

```
      PROGRAM SDVCTERR300OBS_LQR
C*************************************************************************
C
C     PROGRAM SDVCTERR300OBS_LQR.FOR
C
C     WRITTEN BY LT TODD D. HAWKINSON, USN
C
C     FOR AUV THESIS WORK
C
C     THIS PROGRAM CONTROLS A 2 RUDDER SDV WITH A SLIDING MODE
C     CROSS-TRACK-ERROR CONTROLLER FOR STEERING.
C
C     SPEED IS ALSO CONTROLLED USING A SLIDING MODE CONTROLLER
C     DEVELOPED BY LIENARD.
C
C     THE EQUATIONS HAVE BEEN LINEARIZED AT U = 3.0 FT/SEC TO OBTAIN
C     THE SLIDING MODE CONTROL LAWS.
C
C     THE OUTPUT OF THE PROGRAM IS WRITTEN TO FILES, CTECURR.DAT,
C     CTEWAYPT.DAT, LOSALLOUT.DAT.  THESE FILES ARE THEN ACCESSED
C     BY THE DISSPLA PLOTTING PROGRAM  PLOT10.FOR, TO GENERATE
C     OUTPUT GRAPHICS.
C
C     THIS PROGRAM UTILIZES SUBROUTINES FROM THE IMSL MATH
C     LIBRARY, VERSION 1.1, (COPYRIGHT JANUARY 1989 BY IMSL, INC.).
C
C     THE PROGRAM HAS 3 OBSERVERS FOR V, VC, AND UC.
C
C*************************************************************************
C
C  YAW CHARACTERISTICS
C
      REAL NR,NRDOT,NV,NVDOT,NDBR,NDSR,CDY,R,RDOT,PSI,PSIDOT,
     &     PSIX
C
C  SWAY CHARACTERISTICS
C
      REAL YR,YRDOT,YV,YVDOT,V,VDOT,YDBR,YDSR
C
C  SURGE CHARACTERISTICS
C
```

```
      REAL U,UMAX,UD,N,RE,ETA,CD0,XPROP,XUDOT,XVR,XRR,XRDBR,
     &     XRDSR,XVV,XVDBR,XVDSR,XDBRDBR,XDSRDSR,SIGNU,UDOT
C
C  RUDDER CHARACTERISTICS
C
      REAL DMAX,DBR,DSR
C
C  SLIDING MODE CONTROL VARIABLES
C
      REAL SI,EITA,SS1,SS2,SAT1,SAT2,SS3,SAT3
      REAL K11,K12,K13,K14,K15,K16,K21,K22,K23,K24,K25,K26
      REAL S11,S12,S13,S14,S21,S22,S23,S24
C
C  VEHICLE CHARACTERISTICS
C
      REAL WT,L,XG,YG,IZ,RHO,G,M,X(18),H(18),NU
C
C  NAVIGATOR VARIABLES
C
      REAL UCO,UC,VCO,VC
      REAL TARGET,DAWAY,ALPHA
      REAL XD,XD1,XD2,XPOS,XCURR,XCTE,YD,YD1,YD2,YPOS,YCURR,YCTE
      REAL XDOT,YDOT
      INTEGER IWAY,INAV
C
C  OBSERVER VARIABLES
C
      REAL A11,A12,A21,A22,B11,B12,B21,B22,L12,L23,L34,VHAT,UCHAT,
     &     VCHAT,UCOHAT,VCOHAT,Z1,Z2,Z3,Z1DOT,Z2DOT,Z3DOT,S1,S2,S3
C
C  PROGRAM VARIABLES
C
      REAL MM(3,3),MMINV(3,3),F(3),OUT(3)
      REAL DELT
      REAL PIE
      REAL SGN
      REAL VEC1(18),VEC2(18)
      REAL DRAGDX,DRAGXDX,UCF
      REAL TIME,ARG1,ARG2,TALPHA
      INTEGER I,ITER,ISCREEN,K,IOUT
C
C  YAW HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(NR=   -1.640E-2, NRDOT=-3.400E-3, NV=   -7.420E-3,
     &          NVDOT= 1.240E-3, NDBR=  1.290E-2, NDSR= -1.290E-2,
     &          CDY=   3.500E-1)
C
C  SWAY HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(YR=    2.970E-2, YRDOT= 1.240E-3, YV=   -9.310E-2,
```

```
     &                YVDOT=-5.550E-2, YDBR=  2.730E-2, YDSR=  2.730E-2)
C
C  SURGE HYDRODYNAMIC COEFFICIENTS
C
      PARAMETER(UMAX=   6.0    , XUDOT=  -7.580E-3, XVR=   1.890E-2,
     &                XRR=    4.010E-3, XRDBR=   8.180E-4, XRDSR=-8.180E-4,
     &                XVV=    5.290E-2, XVDBR=   1.730E-3, XVDSR= 1.730E-3,
     &                XDBRDBR=-1.010E-2, XDSRDSR=-1.010E-2)
C
C  SLIDING MODE CONTROL VARIABLES
C
      PARAMETER(S11= 1.00000,S12= 0.00000,S13= 0.09200,S14=0.03930,
     &                S21= 0.00000,S22= 1.00000,S23= 1.24230,S24=0.09200,
     &                K11= -0.6207,K12= -7.1479,K13= -3.8547,K14= 0.0000,
     &                K15= -4.4499,K16=-12.0714,
     &                K21=  1.5414,K22= 17.3469,K23=  3.2066,K24=  0.0000,
     &                K25= -4.6361,K26= 13.6038)
C
C  RUDDER CHARACTERISTICS
C
      PARAMETER(DMAX=0.4)
C
C  VEHICLE CHARACTERISTICS
C
      PARAMETER(WT=12000., L=17.425, XG=0.0, YG=0.0, IZ=10000.,
     &                RHO=1.940, G=32.2,   NU=8.47E-4)
C
C  NAVIGATOR VARIABLES
C
      PARAMETER(TARGET=121.975)
C
C  OBSERVER VARIABLES
C
      PARAMETER(A11=-0.042174797, A12=-0.351578237,
     &                A21=-0.002794897, A22=-0.098415870,
     &                B11= 0.012974539, B12= 0.011513052,
     &                B21= 0.004421618, B22=-0.004244119,
     &                S1= -1.0,       S2= -1.1,
     &                S3= -1.2)
C
C  PROGRAM VARIABLES
C
      PARAMETER(PIE=3.141593,DELT=0.01,ITER=50000)
C
C  DEFINE LENGTH X AND HEIGHT TERMS FOR THE INTEGRATION
C
      X(1)=-105.9/12.
      X(2)=-104.3/12.
      X(3)=-99.3/12.
      X(4)=-94.3/12.
```

```
      X(5)=-87.3/12.
      X(6)=-76.8/12.
      X(7)=-66.3/12.
      X(8)=-55.8/12.
      X(9)=72.7/12.
      X(10)=79.2/12.
      X(11)=83.2/12.
      X(12)=87.2/12.
      X(13)=91.2/12.
      X(14)=95.2/12.
      X(15)=99.2/12.
      X(16)=101.2/12.
      X(17)=102.1/12.
      X(18)=103.2/12.
      H(1)=0.0/12.
      H(2)=2.28/12.
      H(3)=8.24/12.
      H(4)=13.96/12.
      H(5)=19.76/12.
      H(6)=25.10/12.
      H(7)=29.36/12.
      H(8)=31.85/12.
      H(9)=31.85/12.
      H(10)=30.00/12.
      H(11)=27.84/12.
      H(12)=25.12/12.
      H(13)=21.44/12.
      H(14)=17.12/12.
      H(15)=12.00/12.
      H(16)=9.12/12.
      H(17)=6.72/12.
      H(18)=0.00/12.
C
C  DATA ENTRY
C
      WRITE(*,*)
10    WRITE(*,*) 'ENTER ORDERED SPEED IN FT/SEC, UD = ?'
      READ(*,*) UD
      IF (UD .LT. 0.0 .OR. UD .GT. UMAX) THEN
            WRITE(*,*)
            WRITE(*,*) ' UD MUST BE:  0.0 <= UD <= 6.0'
            WRITE(*,*) ' RE-ENTER REALISTIC SPEED . . .'
            WRITE(*,*)
            GO TO 10
      ENDIF
      WRITE(*,*)
      WRITE(*,*) 'ENTER GLOBAL CURRENTS, UCO,VCO = ?'
      READ(*,*) UCO,VCO
      OPEN(UNIT=19,FILE='CTECURR.DAT',STATUS='NEW')
      WRITE(19,25) UCO,VCO
```

```
25    FORMAT(F6.3,1X,F6.3)
      CLOSE(UNIT=19)
      WRITE(*,*)
30    WRITE(*,*) 'ENTER CONTROLLER SI ='
      READ(*,*) SI
      WRITE(*,*)
40    WRITE(*,*) 'ENTER CONTROLLER EITA = '
      READ(*,*) EITA
      IF (EITA .LE. 0.0) THEN
            WRITE(*,*)
            WRITE(*,*) 'EITA MUST BE: EITA > 0.0'
            WRITE(*,*) 'RE-ENTER REALISTIC EITA'
            WRITE(*,*)
            GO TO 40
      ENDIF
      WRITE(*,*)
      WRITE(*,*) 'INITIAL HEADING IS 000 DEGREES (DUE NORTH)'
      WRITE(*,*) 'INITIAL POSITION IS (0.0,0.0)'
      WRITE(*,*) 'ENTER FIRST WAY-POINT:  X,Y = ?'
      READ(*,*) XD,YD
C
C  SET INITIAL STARTING POSITION
C
      XD1=0.0
      YD1=0.0
      XD2=XD
      YD2=YD
      OPEN(UNIT=20,FILE='CTEWAYPT.DAT',STATUS='NEW')
      WRITE(20,*) XD1,YD1
      WRITE(20,*) XD2,YD2
C
C  CALCULATE THE MASS
C
      M=WT/G
C
C  SET INITIAL VALUES
C
      ALPHA=ATAN2((YD2-YD1),(XD2-XD1))
      UC=UCO*COS(ALPHA)+VCO*SIN(ALPHA)
      VC=VCO*COS(ALPHA)-UCO*SIN(ALPHA)
      U=UD
      N=UD/0.012
      DBR=0.0
      DSR=0.0
      V=0.0
      R=0.0
      PSI=0.0
      XPOS=0.0
      YPOS=0.0
      XCURR=XPOS
```

```
      YCURR=YPOS
      XCTE=0.0
      YCTE=0.0
      TIME=0.0
      VHAT=0.0
      UCHAT=0.0
      VCHAT=0.0
      UCOHAT=0.0
      VCOHAT=0.0
      Z1=0.0
      Z2=0.0
      Z3=0.0
      Z1DOT=0.0
      Z2DOT=0.0
      Z3DOT=0.0
C
C  CALCULATE THE MASS MATRIX
C
      MM(1,1)=(M-0.5*RHO*L*L*L*XUDOT)
      MM(1,2)=0.0
      MM(1,3)=-M*YG
      MM(2,1)=0.0
      MM(2,2)=(M-0.5*RHO*L*L*L*YVDOT)
      MM(2,3)=(M*XG-0.5*RHO*L*L*L*L*YRDOT)
      MM(3,1)=-M*YG
      MM(3,2)=(M*XG-0.5*RHO*L*L*L*L*NVDOT)
      MM(3,3)=(IZ-0.5*RHO*L*L*L*L*L*NRDOT)
C
C  INVERT THE MASS MATRIX USING IMSL LIBRARY SUBROUTINE
C
      CALL LINRG(3,MM,3,MMINV,3)
C
C  INITIALIZE THE COUNTERS
C
      ISCREEN=1
      INAV=0
      IWAY=1
      IOUT=1
C
C  OPEN OUTPUT FILES
C
      OPEN(UNIT=18,FILE='CTEALLOUT.DAT',STATUS='NEW')
      WRITE(18,*) TIME,XPOS,YPOS,PSI,DBR,DSR,UD,U,V,VHAT,R,UC,UCHAT,
     &            VC,VCHAT
C
C  SIMULATION BEGINS
C
      DO 1000 I=1,ITER
C
C          CALCULATE THE DRAG FORCES
```

```
C
      DO 50 K=1,18
        UCF=(V+R*X(K))
        SGN=1.0
        IF (UCF .LT. 0.0) SGN=-1.0
        VEC1(K)=H(K)*UCF*UCF*SGN
        VEC2(K)=H(K)*UCF*UCF*SGN*X(K)
50    CONTINUE
      CALL TRAP(18,VEC1,X,DRAGDX)
      CALL TRAP(18,VEC2,X,DRAGXDX)
C
C     RHS OF SURGE EQUATION
C
      SIGNU=1.0
      IF ( (U .LT. 0.0) .AND. (N .GT. 0.0) ) SIGNU=-1.0
      IF ( (U .GT. 0.0) .AND. (N .LT. 0.0) ) SIGNU=-1.0
      RE=U*L/NU
      CD0=0.00385 + 1.296E-17*(RE-1.2E7)*(RE-1.2E7)
      F(1)=0.5*RHO*L*L*U*V*(XVDBR*DBR + XVDSR*DSR)
     &       + 0.5*RHO*L*L*L*U*R*(XRDBR*DBR + XRDSR*DSR)
     &       + 0.5*RHO*L*L*XVV*V*V
     &       + (M*XG + 0.5*RHO*L*L*L*L*XRR)*R*R
     &       + (M + 0.5*RHO*L*L*L*XVR)*V*R
     &       + 0.5*RHO*L*L*U*U*(XDBRDBR*DBR*DBR + XDSRDSR*DSR*DSR)
     &       + 0.5*RHO*L*L*CD0*(SIGNU*0.012*0.012*N*N - U*U)
C
C     RHS OF SWAY EQUATION
C
      F(2)=0.5*RHO*L*L*YV*U*V
     &       + (0.5*RHO*L*L*L*YR - M)*U*R
     &       - M*YG*R*R
     &       + 0.5*RHO*L*L*U*U*(YDBR*DBR + YDSR*DSR)
     &       - 0.5*RHO*CDY*DRAGDX
C
C     RHS OF YAW EQUATION
C
      F(3)=0.5*RHO*L*L*L*NV*U*V
     &       + (0.5*RHO*L*L*L*L*NR*U - M*XG*U)*R
     &       - M*YG*V*R
     &       + 0.5*RHO*L*L*L*U*U*(NDBR*DBR + NDSR*DSR)
     &       - 0.5*RHO*CDY*DRAGXDX
C
C     DEFINITION
C
      PSIDOT=R
C
C     MULTIPLY INVERTED MASS MATRIX AND F VECTOR
C
      CALL MURRV(3,3,MMINV,3,3,F,1,3,OUT)
      UDOT=OUT(1)
```

```
      VDOT=OUT(2)
      RDOT=OUT(3)
      XDOT=UCO+U*COS(PSI)-V*SIN(PSI)
      YDOT=VCO+U*SIN(PSI)+V*COS(PSI)
C
C     FIRST ORDER INTEGRATION
C
      U=U + DELT*UDOT
      V=V + DELT*VDOT
      R=R + DELT*RDOT
      PSI=PSI + DELT*PSIDOT
C
C     MAKE PSI TO BE:  -180 < PSI <= 180 DEGREES
C
      IF (PSI .GT. PIE) THEN
          PSI=PSI - 2.0*PIE
      ELSEIF (PSI .LE. -PIE) THEN
          PSI=PSI + 2.0*PIE
      ENDIF
      XPOS=XPOS+DELT*XDOT
      YPOS=YPOS+DELT*YDOT
C
C     CHECK IF TIME FOR NAV UPDATE
C
      INAV=INAV+1
      IF (INAV .NE. 1) GO TO 80
      INAV=0
      XCURR=XPOS
      YCURR=YPOS
      DAWAY=SQRT((XCURR-XD)**2 + (YCURR-YD)**2)
C
C     CHECK DISTANCE TO WAY-POINT
C
      IF (DAWAY .LT. TARGET) THEN
C
C         MISS DISTANCE TO SCREEN, ENTER NEW WAY-POINT,
C         CALCULATE NEW ALPHA,UCHAT,VCHAT,XCTE,YCTE,Z2,Z3
C
          WRITE(*,*)
          WRITE(*,60) IWAY
60        FORMAT(' WAY-POINT #',I1,' REACHED.')
          WRITE(*,65) DAWAY
65        FORMAT(' DAWAY = ',F8.3,' FEET')
          UCOHAT=UCHAT*COS(ALPHA)-VCHAT*SIN(ALPHA)
          VCOHAT=VCHAT*COS(ALPHA)+UCHAT*SIN(ALPHA)
          IWAY=IWAY+1
          WRITE(*,70) IWAY
70        FORMAT(' ENTER WAY-POINT #',I1,':  X,Y = ?')
          READ(*,*) XD,YD
          XD1=XD2
```

```
                YD1=YD2
                XD2=XD
                YD2=YD
                WRITE(20,*) XD2,YD2
                ALPHA=ATAN2((YD2-YD1),(XD2-XD1))
                UC=UCO*COS(ALPHA)+VCO*SIN(ALPHA)
                VC=VCO*C(   ALPHA)-UCO*SIN(ALPHA)
                UCHAT=UCOHAT*COS(ALPHA)+VCOHAT*SIN(ALPHA)
                VCHAT=VCOHAT*COS(ALPHA)-UCOHAT*SIN(ALPHA)
                XCTE=(YCURR-YD1)*SIN(ALPHA)
     &               +(XCURR-XD1)*COS(ALPHA)
                YCTE=(YCURR-YD1)*COS(ALPHA)
     &               -(XCURR-XD1)*SIN(ALPHA)
                Z2=VCHAT+S2*YCTE
                Z3=UCHAT+S3*XCTE+U*COS(PSI-ALPHA)
            ENDIF
C
C           CALCULATE NEW XCTE,YCTE DUE TO CHANGE OF XD,YD AND/OR
C           XCURR,YCURR
C
80          XCTE=(YCURR-YD1)*SIN(ALPHA)+(XCURR-XD1)*COS(ALPHA)
            YCTE=(YCURR-YD1)*COS(ALPHA)-(XCURR-XD1)*SIN(ALPHA)
C*******************************************************************************
C           OBSERVERS FOR SWAY VELOCITY AND CURRENT VELOCITIES
C*******************************************************************************
            L12=(A11*U-S1)/(A21*U)
            L23=-S2
            Z1DOT=S1*Z1 + (A12*U-L12*A22*U+S1*L12)*R
     &           + (B11-B21*L12)*U*U*DBR + (B12-B22*L12)*U*U*DSR
            Z2DOT=S2*Z1 + S2*Z2 - L23*U*SIN(PSI-ALPHA) + S2*L12*R
     &           + S2*L23*YCTE
            Z3DOT=S3*Z3 - S3*S3*XCTE
C
C           FIRST ORDER INTEGRATION
C
            Z1=Z1+DELT*Z1DOT
            Z2=Z2+DELT*Z2DOT
            Z3=Z3+DELT*Z3DOT
C
C           FINAL OBSERVER EQUATIONS
C
            VHAT=L12*R + Z1
            VCHAT=L23*YCTE + Z2
            UCHAT=Z3 - S3*XCTE - U*COS(PSI-ALPHA)
C*******************************************************************************
C           END OBSERVERS
C*******************************************************************************
C
C           CALCULATE THE CONTROL LAWS
C
```

```
      IF (VC/U .GT. 1.0) THEN
           ARG1=1.0
      ELSEIF (VC/U .LT. -1.0) THEN
           ARG1=-1.0
      ELSE
           ARG1=VC/U
      ENDIF
      IF ((PSI .GE. 0.0 .AND. PSI .LE. PIE) .AND.
     &    (ALPHA .GE. -PIE .AND. ALPHA .LE. PSI-PIE)) THEN
           TALPHA=ALPHA + 2.0*PIE
           ARG2=PSI-TALPHA
      ELSEIF ((PSI .GE. -PIE .AND. PSI .LT. 0.0) .AND.
     &        (ALPHA .GT. PSI+PIE .AND. ALPHA .LE. PIE)) THEN
           TALPHA=ALPHA - 2.0*PIE
           ARG2=PSI-TALPHA
      ELSE
           TALPHA=ALPHA
           ARG2=PSI-TALPHA
      ENDIF
      SS1=S11*VHAT+S12*R+S13*ARG2+S14*YCTE+
     &    ((SI/EITA)*((K13*K26-K16*K23)/(K15*K26-K16*K25))+S13)*
     &    ASIN(ARG1)
      SS2=S21*VHAT+S22*R+S23*ARG2+S24*YCTE+
     &    ((SI/EITA)*((K15*K23-K13*K25)/(K15*K26-K16*K25))+S23)*
     &    ASIN(ARG1)
      SS3=U-UD
      IF (ABS(SS1) .LT. SI) SAT1=SS1/SI
      IF (SS1 .LE. -SI) SAT1=-1.0
      IF (SS1 .GE. SI) SAT1=1.0
      IF (ABS(SS2) .LT. SI) SAT2=SS2/SI
      IF (SS2 .LE. -SI) SAT2=-1.0
      IF (SS2 .GE. SI) SAT2=1.0
      IF (ABS(SS3) .LT. SI) SAT3=SS3/SI
      IF (SS3 .LE. -1.0) SAT3=-1.0
      IF (SS3 .GE. 1.0) SAT3=1.0
      DBR=K11*VHAT + K12*R + K13*ARG2 + K14*YCTE
     &    + EITA*K15*SAT1 + EITA*K16*SAT2
      DSR=K21*VHAT + K22*R + K23*ARG2 + K24*YCTE
     &    + EITA*K25*SAT1 + EITA*K26*SAT2
      N=-1153.9*SAT3 + 83.33*U
C
C     CHECK FOR SATURATION
C
      IF (DBR .GT. DMAX) DBR=DMAX
      IF (DBR .LT. -DMAX) DBR=-DMAX
      IF (DSR .GT. DMAX) DSR=DMAX
      IF (DSR .LT. -DMAX) DSR=-DMAX
      IF (N .GE. 500.0) N=500.0
      IF (N .LE. -500.0) N=-500.0
C
```

```
C          OUTPUT
C
           IF (PSI  .LT. 0.0 .AND. PSI  .GT. -PIE) THEN
                PSIX=(PSI+2.0*PIE)*180.0/PIE
           ELSE
                PSIX=PSI*180.0/PIE
           ENDIF
           IF (ISCREEN .EQ. 200) THEN
                TIME=I/100.
                WRITE(*,90) TIME,PSIX,XPOS,YPOS
90              FORMAT(' TIME= ',F7.2,2X,'PSI= ',F9.4,2X,
     &                  'XPOS= ',F8.2,2X,'YPOS= ',F8.2)
                ISCREEN=0
           ENDIF
           IF (IOUT .EQ. 20) THEN
                TIME=I/100.
                WRITE(18,*) TIME,XPOS,YPOS,PSIX,DBR,DSR,UD,U,
     &                      V,VHAT,R,UC,UCHAT,VC,VCHAT
                IOUT=0
           ENDIF
           ISCREEN=ISCREEN+1
           IOUT=IOUT+1
1000  CONTINUE
      CLOSE(UNIT=18)
      CLOSE(UNIT=20)
      STOP
      END
C
C  NUMERICAL INTEGRATION ROUTINE USING THE TRAPEZOIDAL RULE
C
      SUBROUTINE TRAP(N,A,B,OUT)
      INTEGER N,N1,I
      REAL A(N),B(N),OUT,OUT1
      N1=N-1
      OUT=0.0
      DO 10 I=1,N1
           OUT1 = 0.5*(A(I) + A(I+1))  *  (B(I+1) - B(I))
      OUT = OUT + OUT1
10    CONTINUE
      RETURN
      END
```

LIST OF REFERENCES

1. Utkin, Vadim I., "Variable Structure Systems with Sliding Modes," *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 2, April 1977.

2. Slotine, Jean-Jacques E., *Tracking Control of Nonlinear Systems Using Sliding Surfaces*, Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1983.

3. Lienard, David E., *Autopilot Design for Autonomous Underwater Vehicles Based on Sliding Mode Control*, Master's Thesis, Naval Postgraduate School, June 1990.

4. Sur, Joo-No, *Design and Investigation of a Dive Plane Sliding Mode Compensator for an Autonomous Underwater Vehicle*, Master's Thesis, Naval Postgraduate School, September 1989.

5. Papoulias, F.A., Cristi, R., Marco, D., and Healey, A.J., "Modeling, Sliding Mode Control Design, and Visual Simulation of AUV Dive Plane Dynamic Response," paper presented at the 6th Annual International Symposium on Unmanned Untethered Submersible Technology, Washington, D.C., June 1989.

6. Utkin, V.I. and Yang, K.D., "Methods for Constructing Discontinuity Planes in Multidimensional Variable Structure Systems," *Automatic Remote Control*, Vol. 32, No. 12, October 1978.

7. Naval Coastal Systems Center Technical Memorandum 231-78, *SDV Simulator Hydrodynamic Coefficients*, by Smith, N.C., Crane, J.W., and Summey, D.C., Unclassified, June 1978.

8. Saunders, Thomas E., *Performance of Small Thrusters and Propulsion Systems*, Master's Thesis, Naval Postgraduate School, March 1990.

9. Norrby, R.Å. and Ridley, D.E., "Notes on Thrusters for Ship Maneuvering and Dynamic Positioning," *Society of Naval Architects and Marine Engineers*, Vol. 88, pp. 377-402, 1980.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code MEHy Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Dr. G. Dobeck, Code 4210 Naval Coastal Systems Command Panama City, FL 32407-5000	1
5.	Dr. Richard Nadolink Head, Weapon Technology Division Chair in Hydrodynamics Naval Underwater Systems Center Newport, RI 02841-5047	1
6.	RADM Evans, Code SEA-92 Naval Sea Systems Command Washington, DC 20362	1
7.	Dan Steiger Marine Systems Group Naval Research Laboratory Washington, DC 20032	1
8.	Technical Library Branch, Code E23 Naval Surface Warfare Center Silver Spring, MD 20903-5000	1

9. Naval Engineering Curricular Office, Code 34 — 1
Naval Postgraduate School
Monterey, CA 93943-5000

10. Dr. Dana Yoerger — 1
Woods Hole Oceanographic Institute
Woods Hole, MA 02543

11. LT Todd D. Hawkinson — 1
Pearl Harbor Naval Shipyard
Pearl Harbor, HI 96860-5350

12. Professor F. A. Papoulias, Code MEPa — 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93943-5000

13. Professor R. Christi, Code 62Cx — 1
Department of Electrical & Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5000